



Sistemas Informáticos

Curso 2007-2008

FVE Project Manager:

**Gestión de procesos software y control
de versiones sobre dotProject.**

Ignacio Estirado Casado

Íñigo Fábregas Martínez

Alejandro Vilela Palacios

Dirigido por:

Jorge Gómez Sanz

Departamento de Ingeniería del Software e
Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid

Nosotros, Ignacio Estirado Casado, Íñigo Fábregas Martínez y Alejandro Vilela Palacios, creadores del presente documento y del proyecto de Sistemas Informáticos "*FVE Project Manager: Gestión de procesos software y control de versiones sobre dotProject*", autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Ignacio Estirado Casado
DNI: 70801323-D

**Íñigo Fábregas
Martínez**
DNI: 18054768-K

**Alejandro Vilela
Palacios**
DNI: 11838386-X

A Thomas Wally, Nicolae Morot, y tantos otros.

Y en especial a los que me enseñaron a pelear.

(Ignacio Estirado Casado)

A mi familia y a mis amigos que siempre están tras de mí.

A mis compañeros de proyecto Alex y Nacho

por haberme hecho participe del mismo.

(Íñigo Fábregas Martínez)

A mis padres, que siempre me dieron la oportunidad de cumplir mis sueños.

Y a quienes me ayudaron a ponerme en pie cuando había caído.

(Alejandro Vilela Palacios)

*A Jorge J. Gómez Sanz por darnos la oportunidad de desarrollar este
proyecto.*

(Común)

<<I live in the real world. I'm not a big believer in magic. But this place is different. It's special. The others don't want to talk about it because it scares them. But we all know it. We all feel it. Is your white rabbit a hallucination? Probably. But what if everything that happened here, happened for a reason?>>

John Locke

Resumen en castellano

Nuestro proyecto tiene como finalidad aportar una aplicación libre y de código abierto que permita asistir en el desarrollo de un proyecto software facilitando la elección de un proceso de desarrollo; su ejecución entendida como la obtención de los resultados esperables de un proceso de desarrollo; y el acceso al mismo por parte de los participantes elegidos. Esta aplicación se llama FVE Project Manager.

Existen algunas soluciones libres, pero distan mucho de ser completas, sobre todo por no estar integradas con sistemas de control de versiones. Así, en lugar de iniciar un proyecto desde cero, decidimos modificar un gestor de proyectos existente, dotProject, ampliando sus características.

Con nuestra herramienta se pueden crear nuevas definiciones de procesos o importarlas en formato XML desde EPFComposer. Se permite la creación de productos de trabajo, gestión de los roles del proceso y personalización de las tareas. Una vez instanciado el proceso, los participantes pueden empezar a trabajar en el proyecto con todos los elementos definidos en el mismo mediante una intuitiva interfaz gráfica que brinda la aplicación.

Otro logro notable es el de integrar un sistema de control de cambios (SVN). Ahora mismo, no existe ninguna extensión de dotproject para soportar SVN. Este servicio se integra de tal forma que no hay necesidad de que el cliente instale ningún tipo de programa adicional en su ordenador, pudiendo además acceder a este repositorio de forma independiente a dotProject.

El sistema ha sido desarrollado en los lenguajes PHP, JSP, Java y Javascript y requiere para su despliegue de un servidor con servicios web compatibles con PHP, un servidor de base de datos MySQL, un servidor de aplicaciones para J2EE y la instalación de los servicios de Subversion en el servidor.

Todas las herramientas utilizadas para el desarrollo son Open Source, por lo que podemos devolver a la comunidad de desarrolladores el beneficio de poder usar estas herramientas con una aplicación libre que ofrece servicios disponibles generalmente bajo pago.

La agrupación de todas estas características sobre un gestor de proyectos libre como dotProject supone la creación de una funcionalidad no existente en aplicaciones libres y da respuesta a un gran número de solicitudes por parte de la comunidad de usuarios.

Summary in English

The purpose of our project is contributing a free and Open Source application that allows the administration of a project management entire cycle by the supplying of new functionalities that provide more power and flexibility to the execution of these projects. The name of this application is FVE Project Manager.

There are several free solutions, but they are far from being complete, especially for not being integrated with version control systems. So, instead of starting a project from scratch, we decided to modify an existing project manager, dotproject, expanding its features.

With our tool it is possible to create new process definitions or import them in XML format from EPFComposer. It allows the creation of workproducts, process roles management and personalization of the process of tasks. Once the process is instantiated, the participants may start working on it with all the defined elements in it through an intuitive graphical interface that the application provides.

Another remarkable achievement is integrating a version control system (SVN). Right now, there is no extension of dotproject that works with SVN. This service is integrated so that there is no need for the customer to install any additional software on his computer. He can also access this repository independently from dotproject.

The system has been developed in PHP, JSP, Java and Javascript languages. It requires for its deployment a PHP compatible web services server, a MySQL database server, a J2EE application server and the installation of the Subversion services in the server.

All the tools that we used in the development are Open Source, so we can return to the developer community the benefit of being able to use these tools with a free application that provides non-free services.

The grouping of all these features on a free project manager as dotproject involves creating a non-existent feature in free applications and responded to numerous requests by the user community.

Palabras Clave

Palabras clave (Castellano):

Gestión de proyectos, Control de versiones, Procesos de desarrollo, Procesos software, dotProject, Subversion, Eclipse Process Framework, Código abierto, Ingeniería del Software.

Key references (English):

Project Management, Version control, Delivery Process, Software process, dotProject, Subversion, Eclipse Process Framework, Open Source, Software Engineer.

Índice

A continuación se muestra el índice del presente documento:

Resumen en castellano	Pág. 7.
Summary in English	Pág. 9.
Palabras clave	Pág. 11.
1. Introducción	Pág. 28.
1.1. Estructura de la memoria	Pág. 29.
2. Herramientas implicadas	Pág. 31.
2.1 Gestores de proyectos	Pág. 31.
2.1.1 Función	Pág. 31.
2.1.2 Justificación de aplicación en el proyecto	Pág. 32.
2.1.3 Estudio de gestores	Pág. 33.
2.1.4 Elección de gestor: dotProject	Pág. 33.
2.2. EPF: Gestión de procesos de desarrollo	Pág. 35.
2.2.1. Eclipse Process Framework	Pág. 35.
2.2.2. Comparación con Rational Method Composer: Similitudes y Carencias	Pág. 36.
2.2.3. Elementos de EPF Composer	Pág. 37.
2.2.4. Uso de EPF Composer	Pág. 38.
2.2.5. Justificación de aplicación en el proyecto	Pág. 38.
2.2.6. Definición en detalle de una metodología (aplicable a proyectos de Ingeniería del Software a partir de la experiencia de los integrantes del grupo) usando EPF Composer	Pág. 39.
2.2.6.1 - Definición de iteraciones	Pág. 40.
2.2.6.2 - Definición de tareas	Pág. 41.
2.2.6.3 - Definición de roles	Pág. 42.
2.2.6.4 - Definición de productos de trabajo	Pág. 43.
2.2.6.5 - Exportación XML	Pág. 46.
2.3. Repositorio de control de versiones	Pág. 47.
2.3.1. Justificación de aplicación en el proyecto	Pág. 47.

2.3.2. CVS vs SVN: Elección de Subversion.....	Pág. 47.
2.3.3. Gestor SVN: Necesidad de herramienta intermedia	Pág. 48.
2.3.4. Elección de SVNKit: Ventajas y desventajas	Pág. 49.
3. Análisis	Pág. 50.
3.1. Requisitos funcionales.....	Pág. 50.
3.2. Requisitos no funcionales	Pág. 53.
3.3. Casos de uso	Pág. 58.
3.3.1. Creación de un proyecto FVE	Pág. 61.
3.3.2. Ejecución de un proyecto FVE	Pág. 67.
3.3.3. Modificación de un proyecto FVE.....	Pág. 92.
3.3.4. Modificación de los parámetros del módulo de Subversion....	Pág. 119.
4. Desarrollo	Pág. 125.
4.1. Dónde termina dotProject y empieza FVE.....	Pág. 125.
4.1.1. Qué funcionalidades ofrece dotProject	Pág. 125.
4.1.2. Diferencias entre elementos de dotProject y de FVE	Pág. 126.
4.1.3. Dónde se modifica el código base de dotProject	Pág. 127.
4.2. Limitaciones y convenios a seguir de dotProject.....	Pág. 128.
4.3. Arquitectura del sistema	Pág. 129.
4.3.1. Diagramas de despliegue	Pág. 129.
4.3.1.1. Despliegue en LAN (Local Area Network) con Base de Datos Integrada	Pág. 130.
4.3.1.2. Despliegue en LAN (Local Area Network) con Base de Datos Separada	Pág. 130.
4.3.1.3. Despliegue en WAN (Wide Area Network) con Base de Datos Separada	Pág. 131.
4.3.2. Diseño del sistema	Pág. 132.
4.3.2.1. Mapa y funcionamiento estructural de la aplicación	Pág. 133.
4.3.2.2. Clases utilizadas	Pág. 139.
4.4. Estructura de BBDD.....	Pág. 145.
4.4.1. Dependencia de dotProject	Pág. 145.

4.4.2. Esquema Entidad – Relación	Pág. 145.
4.4.3. Explicación de la estructura	Pág. 147.
4.4.4. Listado de las tablas utilizadas	Pág. 148.
4.4.4.1. Tablas que se utilizan de dotProject	Pág. 148.
4.4.4.1.1. – projects	Pág. 148.
4.4.4.1.2. – tasks	Pág. 149.
4.4.4.1.3. – users	Pág. 150.
4.4.4.2. Tablas nuevas	Pág. 151.
4.4.4.2.1. - fve_projects.....	Pág. 151.
4.4.4.2.2. - fve_project_and_workers.....	Pág. 152.
4.4.4.2.3. - fve_tasks.....	Pág. 152.
4.4.4.2.4. - fve_workproducts	Pág. 153.
4.4.4.2.5. - fve_roles	Pág. 154.
4.4.4.2.6. - fve_user_and_possible_roles	Pág. 154.
4.4.4.2.7. - fve_role_primary_task.....	Pág. 155.
4.4.4.2.8. - fve_role_additional_task	Pág. 155.
4.4.4.2.9. - fve_role_responsible_workproduct	Pág. 156.
4.4.4.2.10. - fve_role_modifies_workproduct	Pág. 157.
4.4.4.2.11. - fve_user_primary_task	Pág. 157.
4.4.4.2.12. - fve_user_with_role_additional_task	Pág. 158.
4.4.4.2.13. - fve_user_responsible_workproduct	Pág. 158.
4.4.4.2.14. - fve_user_with_role_modifies_workproduct	Pág. 159.
4.4.4.2.15. - fve_task_and_workproducts	Pág. 160.
4.4.4.2.16. - fve_user_wizard_sessions	Pág. 160.
4.4.4.2.17. - fve_epf_import_aux	Pág. 161.
4.4.4.2.18. - fve_epf_parent_aux	Pág. 161.
4.4.4.2.19. - fve_epf_dependency_aux.....	Pág. 162.
4.4.4.2.20. - fve_templates	Pág. 163.
4.4.4.2.21. - fve_workproduct_and_template	Pág. 163.

4.4.4.2.22. - fve_parameters	Pág. 164.
4.4.5. Transaccionalidad en la base de datos.....	Pág. 165.
4.5. Estructura del módulo de Subversion	Pág. 166.
4.5.1 Paquete de componentes J2EE	Pág. 166.
4.5.1.1. Paquete de clases java “Operaciones”	Pág. 166.
4.5.1.2. Conjunto de páginas jsp.....	Pág. 168.
4.6 Detalles de implementación	Pág. 169.
4.6.1. Wizard de instanciación de proyecto	Pág. 169.
4.6.2. Importador XML de procesos de desarrollo de EPF Composer	Pág. 171.
4.6.3. Doble funcionalidad en contenidos FVE (en núcleo de dotProject y en módulo FVE)	Pág. 173.
4.6.4. JavaScript dinámicamente generado	Pág. 174.
4.6.5. Subversion.....	Pág. 175.
4.6.5.1. Elección de las herramientas principales y consideraciones técnicas sobre la comunicación dispuesta entre ambas.....	Pág. 175.
4.6.5.2. Estructura física de los ficheros de Subversion.....	Pág. 176.
4.6.5.3. Permisos en Subversion	Pág. 177.
4.6.5.4. Transmisión de los ficheros de Subversion vía web.....	Pág. 178.
4.6.5.5. Visualización del repositorio con la ayuda de árboles XloadTree	Pág. 178.
5. Organización del proyecto	Pág. 180.
5.1 Planificación de proyecto	Pág. 180.
5.1.1 Elaboración	Pág. 180.
5.1.1.1 - Iteración 0: Definición e investigación del proyecto.....	Pág. 180.
5.1.1.2 - Iteración 1: Análisis detallado de herramientas implicadas	Pág. 180.
5.1.1.3 - Iteración 2: Arquitectura y diseño del sistema.....	Pág. 181.
5.1.2 Construcción	Pág. 181.
5.1.2.1 – Iteración 3: Creación y modificación de entidades principales y funciones Subversion	Pág. 181.

5.1.2.2 – Iteración 4: Importación de procesos de desarrollo y asistente de instanciación de proyecto	Pág. 182.
5.1.2.3 – Iteración 5: Gestión completa de entidades e integración de Subversion	Pág. 182.
5.1.3 Depuración y pruebas	Pág. 183.
5.1.3.1 – Iteración 6: Reorganización de contenidos, funcionalidades nuevas relativas a lo planificado y solución de errores	Pág. 183.
5.1.3.2 – Iteración 7: Solución de errores, revisión de documentación y generación de ejemplos.....	Pág. 184.
5.2 Reparto del trabajo	Pág. 185.
6. Ejemplo de uso	Pág. 187.
6.1 Introducción.....	Pág. 187.
6.2 Instanciación de proyecto.....	Pág. 187.
6.3 Visión y trabajo en el proyecto	Pág. 191.
6.4 Visión y trabajo con el repositorio.....	Pág. 197.
6.5. Otros ejemplos	Pág. 201.
6.5.1. OpenUp	Pág. 202.
6.5.2. Scrum	Pág. 202.
6.5.3. eXtreme Programming	Pág. 203.
7. Conclusiones y trabajo futuro.....	Pág. 204.
7.1 Datos de la aplicación resultante	Pág. 204.
7.2. Trabajo futuro	Pág. 206.
8. Bibliografía.....	Pág. 209.
Anexo A. Guía de estilo.....	Pág. 213.
Anexo B. Trabajo relacionado con gestores de proyectos	Pág. 214.
B.1. Estudio de gestores de proyectos	Pág. 214.
B.1.1 – DotProject	Pág. 214.
B.1.2 – EgroupWare	Pág. 215.
B.1.3 – PhpCollab	Pág. 216.
B.1.4 – PhpProjekt	Pág. 217.
B.1.5 - Open Workbench.....	Pág. 218.

B.1.6 - Web2project	Pág. 219.
B.1.7 – ActiveCollab	Pág. 220.
B.1.8 - Gestor de proyecto a bajo nivel, con gestión y seguimiento de control de versiones (Subversion): Trac.....	Pág. 221.
B.2. Introducción y uso general de dotProject	Pág. 223.
Anexo C. Trabajo relacionado con EPF.....	Pág. 225.
C.1. Uso de EPF Composer	Pág. 225.
C.1.1 – Configuración de entorno para un nuevo proceso de desarrollo	Pág. 225.
C.1.2 – Creación de un proceso de desarrollo vacío	Pág. 225.
C.1.3 – Creación de uno o más conjuntos de roles (Roleset), para poder crear posteriormente uno o más roles asociados.	Pág. 225.
C.1.4 – Creación de uno o más tipos de productos de trabajo (Workproduct Kinds), para poder crear posteriormente uno o más Work Products.	Pág. 226.
C.1.5 – Creación de tareas y jerarquías en un proceso de desarrollo: Iteraciones, Fases, Actividades y Tareas.	Pág. 226.
C.1.6 – Asociación de entidades.	Pág. 227.
C.1.7 – Publicación Web del proceso de desarrollo	Pág. 227.
C.1.8 – Exportación de datos	Pág. 228.
C.2. Metodologías definidas y utilizadas en el proyecto.....	Pág. 229.
C.2.1 – OpenUp y OpenUp Basic.....	Pág. 230.
C.2.2 – Scrum	Pág. 231.
C.2.3 – eXtreme Programming	Pág. 232.
Anexo D. Trabajo relacionado con Subversion	Pág. 233.
D.1. Instalación y uso de Subversion.....	Pág. 233.
D.1.1. – Instalación	Pág. 233.
D.1.2. - Creación y uso de un repositorio: Guía rápida	Pág. 233.
D.2. Estudio de librerías de gestión SVN	Pág. 235.
D.2.1. – Librerías de Subversion para PHP	Pág. 235.
D.2.2. - USVN.....	Pág. 235.
D.2.3. - SVNKit.....	Pág. 235.

Anexo E. Prototipos	Pág. 237.
E.1 Prototipo de gestión de tareas y dependencias	Pág. 237.
E.1.1 - Introducción e instalación del prototipo	Pág. 237.
E.1.2 - Página principal	Pág. 237.
E.1.3 - Interacción con proyectos	Pág. 237.
E.1.4 - Tareas avanzadas	Pág. 238.
E.2 Prototipo de interfaz de usuario.....	Pág. 239.
E.2.1 – Introducción	Pág. 239.
E.2.2 – Instalación	Pág. 239.
E.2.3 - Estructura y funcionamiento	Pág. 240.
Anexo F. Manual de instalación (en inglés)	Pág. 245.
F.1. Introduction	Pág. 245.
F.2. Installing the necessary services	Pág. 245.
F.3. Installing the application	Pág. 245.
F.4. Linux installation	Pág. 251.
Anexo G. Manual de usuario (en inglés)	Pág. 253.
G.1. Introduction.....	Pág. 253.
G.2. Creating a new Project	Pág. 255.
G.3. Modifying your Project	Pág. 269.
G.4. Subversion	Pág. 297.
G.5. Glossary.....	Pág. 304.

Índice de Figuras

A continuación se muestra el índice de las imágenes y tablas presentes en este documento:

Eclipse Process Framework logo	Pág. 35.
What is the Eclipse Process Framework: Arquitectura	Pág. 36.
Listado de iteraciones	Pág. 40.
Diagrama de dependencias entre iteraciones.....	Pág. 40.
Diagrama de actividad 1: Propose an idea.	Pág. 41.
Diagrama de actividad 2: Investigate technologies.	Pág. 41.
Diagrama de actividad 3: Publish Project.....	Pág. 42.
Listado de roles	Pág. 42.
Listado de productos de trabajo	Pág. 43.
Role Analyst en detalle.	Pág. 44.
Diagrama de actividad detallado 1: Make a Design.	Pág. 45.
Diagrama de actividad detallado 2: Make a Design.	Pág. 45.
Diagrama de actividad detallado 1: Investigate technologies.	Pág. 46.
Diagrama de actividad detallado 2: Investigate technologies.	Pág. 46.
Diferencias entre CVS y SVN.	Pág. 48.
SVNKit logo.....	Pág. 49.
Diagrama de despliegue. Esquema A.	Pág. 130.
Diagrama de despliegue. Esquema B.	Pág. 131.
Diagrama de despliegue. Esquema C.	Pág. 132.
Comparación entre la estructura de dotProject y la de FVE Project Manager	Pág. 133.
Diagrama de componentes del módulo FVE utilizado por nuestra aplicación	Pág. 134.
Diagrama de componentes del componente EPF ubicado dentro del módulo FVE	Pág. 136.
Menú superior de aplicación.	Pág. 137.

Diagrama de secuencia que explica cómo se llevan a cabo las llamadas entre los archivos para responder a una petición de una página web en FVE Project Manager por parte del usuario.	Pág. 138.
Código relativo a clase index.php	Pág. 139.
Submenú de la aplicación.....	Pág. 139.
Clase SessionFVE.....	Pág. 140.
Secuencia de acciones en la sesión FVE.....	Pág. 141.
Clase SessionWizard.....	Pág. 142.
Clases de acceso a base de datos 1.	Pág. 143.
Clases de acceso a base de datos 2.	Pág. 144.
Clases de acceso a base de datos 3.	Pág. 144.
Clases de acceso a base de datos 4.	Pág. 145.
Esquema Entidad – Relación 1.	Pág. 146.
Esquema Entidad – Relación 2.	Pág. 147.
Representación visual de tabla projects.....	Pág. 149.
Representación visual de tabla tasks.....	Pág. 150.
Representación visual de tabla users	Pág. 151.
Representación visual de tabla fve_projects.....	Pág. 151.
Representación visual de tabla fve_project_and_workers.....	Pág. 152.
Representación visual de tabla fve_tasks.....	Pág. 152.
Representación visual de tabla fve_workproducts	Pág. 153.
Representación visual de tabla fve_roles	Pág. 154.
Representación visual de tabla fve_user_and_possible_roles	Pág. 154.
Representación visual de tabla fve_role_primary_task.....	Pág. 155.
Representación visual de tabla fve_role_additional_task	Pág. 156.
Representación visual de tabla fve_role_responsible_workproduct	Pág. 156.
Representación visual de tabla fve_role_modifies_workproduct	Pág. 157.
Representación visual de tabla fve_user_primary_task.....	Pág. 157.
Representación visual de tabla fve_user_with_role_additional_task.....	Pág. 158.
Representación visual de tabla fve_user_responsible_workproduct.....	Pág. 159.

Representación visual de tabla fve_user_with_role_modifies_workproduct	Pág. 159.
Representación visual de tabla fve_task_and_workproducts	Pág. 160.
Representación visual de tabla fve_user_wizard_sessions	Pág. 160.
Representación visual de tabla fve_epf_import_aux	Pág. 161.
Representación visual de tabla fve_epf_parent_aux	Pág. 162.
Representación visual de tabla fve_epf_dependency_aux.....	Pág. 162.
Representación visual de tabla fve_templates	Pág. 163.
Representación visual de tabla fve_workproduct_and_template	Pág. 164.
Representación visual de tabla fve_parameters	Pág. 164.
Esquema de transaccionalidad.	Pág. 165.
Diagrama de capas incluyendo componentes J2EE	Pág. 166.
Diagrama de componentes del paquete de operaciones de Subversión.	Pág. 168.
Diagrama de componentes del módulo de Subversion	Pág. 169.
Menú superior de pasos en el wizard.....	Pág. 170.
Jerarquía de funciones en importador XML.....	Pág. 172.
Visualización común de entidades en dotProject y fve.....	Pág. 174.
Ejemplo de usuarios en el wizard.	Pág. 174.
Función createArray con javascript dinámico.....	Pág. 175.
Esquema de archivos Passwd y Authz.	Pág. 177.
Código de ejemplo de fichero a leer por XTreeMap.....	Pág. 179.
Representación del árbol de un repositorio de ejemplo.	Pág. 179.
Ejemplo de uso: 1. Instanciación paso 1.....	Pág. 188.
Ejemplo de uso: 2. Instanciación paso 3.....	Pág. 188.
Ejemplo de uso: 3. Instanciación paso 4.....	Pág. 189.
Ejemplo de uso: 4. Instanciación paso 5.....	Pág. 190.
Ejemplo de uso: 5. Instanciación paso 6.....	Pág. 191.
Ejemplo de uso: 6. Ver proyecto	Pág. 192.
Ejemplo de uso: 7. Lista de tareas.....	Pág. 192.

Ejemplo de uso: 8. Diagrama de gantt de una iteración	Pág. 193.
Ejemplo de uso: 9. View Task (dotProject)	Pág. 193.
Ejemplo de uso: 10. View EPF Task	Pág. 194.
Ejemplo de uso: 11. Listado de Roles	Pág. 194.
Ejemplo de uso: 12. Mostrar workproduct RiskList	Pág. 195.
Ejemplo de uso: 13. Pantalla de view user	Pág. 196.
Ejemplo de uso: 14. FVE stats report	Pág. 197.
Ejemplo de uso: 15. Árbol del repositorio	Pág. 197.
Ejemplo de uso: 16. Commit de archivos	Pág. 198.
Ejemplo de uso: 17. Mostrar Plantilla RiskListIS	Pág. 198.
Ejemplo de uso: 18. Información del repositorio	Pág. 199.
Ejemplo de uso: 19. Ejemplo de workproduct del proyecto de Simtraffic visto como fichero físico en el FileManager	Pág. 199.
Ejemplo de uso: 20. Ejemplo de confirmación de borrado de workproduct del proyecto de Simtraffic desde el gestor de archivos FileManager	Pág. 200.
Ejemplo de uso: 21. Situación posterior a la del borrado del único fichero de la carpeta de workproducts en el ejemplo de Simtraffic	Pág. 201.
Tareas definidas en OpenUp e importadas a FVE	Pág. 202.
Tareas definidas en Scrum e importadas a FVE	Pág. 202.
Tareas definidas en XP e importadas a FVE	Pág. 203.
CLOC Report: Líneas de código por lenguaje de la aplicación final.	Pág. 205.
CLOC Report: Líneas de código por lenguaje de la nueva funcionalidad	Pág. 205.
Gráfico de actividad del CVS del proyecto.	Pág. 206.
DotProject logo	Pág. 223.
Interfaz: Comportamiento esperado de dotProject.	Pág. 224.
Diagrama de OpenUp	Pág. 230.
Diagrama de Scrum	Pág. 231.
Diagrama de eXtreme Programming	Pág. 232.
Prototipo 2: Gestión de Workflows	Pág. 240.
Prototipo 2: Visualización de tarea	Pág. 242.

Prototipo 2: Listado de roles y usuarios.....	Pág. 243.
Prototipo 2: Listado de Requisitos.	Pág. 244.
Prototipo 2: Información de un requisito.	Pág. 244.
Manual de Instalación 1: Error: Don't worry, it's not so fatal	Pág. 246.
Manual de Instalación 2: This is the dotproject initial check page	Pág. 246.
Manual de Instalación 3: Database connection configuration page	Pág. 247.
Manual de Instalación 4: The module is installed but not activated yet	Pág. 248.
Manual de Instalación 5: Your FVE module is active and ready to use!	Pág. 248.
Manual de Instalación 6: Follow the links to access configuration pane	Pág. 249.
Manual de Instalación 7: The FVE Subversion Menu access	Pág. 249.
Manual de Instalación 8: Subversion configuration Panel (with default configuration)	Pág. 250.
Manual de Instalación 9: Initial repository creation panel.....	Pág. 250.
Manual de Instalación 10: Congratulations, you have activated your Subversion module correctly!	Pág. 250.
Manual de Instalación 11: Comando svnserve	Pág. 251.
Manual de Instalación 12: FVE logo	Pág. 252.
Manual de Usuario 1: The button that launches the Project Creation Wizard	Pág. 256.
Manual de Usuario 2: These buttons control the flow of the wizard steps	Pág. 256.
Manual de Usuario 3: First step form: filling project main data	Pág. 257.
Manual de Usuario 4: Second step form: importing from an XML File and creating your repository	Pág. 258.
Manual de Usuario 5: Third step: assigning users to your project	Pág. 259.
Manual de Usuario 6: Fourth step: assigning users to available roles ...	Pág. 259.
Manual de Usuario 7: Fifth step: configuring a workproduct	Pág. 260.
Manual de Usuario 8: A workproduct with a template associated to it ..	Pág. 261.
Manual de Usuario 9: Sixth step: configuring a task	Pág. 262.
Manual de Usuario 10: Creating a new project	Pág. 264.

Manual de Usuario 11: Viewing the project info	Pág. 264.
Manual de Usuario 12: Converting your project into a EPF project	Pág. 265.
Manual de Usuario 13: Converting your project into a EPF project: Adding users	Pág. 265.
Manual de Usuario 14: The new EPF info of your project	Pág. 266.
Manual de Usuario 15: The button to create a new task in your project	Pág. 266.
Manual de Usuario 16: The EPF Info of your task is empty	Pág. 267.
Manual de Usuario 17: First step in your FVE task configuration.....	Pág. 267.
Manual de Usuario 18: Second step in your FVE task configuration.....	Pág. 268.
Manual de Usuario 19: The third and last step in your FVE task configuration	Pág. 269.
Manual de Usuario 20: The button that launches the Role Creation Wizard in the button "New role"	Pág. 270.
Manual de Usuario 21: The button that launches the Role Creation Wizard in the tab "EPF Roles"	Pág. 271.
Manual de Usuario 22: The first step in EPF Role Creation Wizard.	Pág. 271.
Manual de Usuario 23: The second step in EPF Role Creation Wizard. Initial aspect.	Pág. 272.
Manual de Usuario 24: The second step in EPF Role Creation Wizard, with Users selected	Pág. 272.
Manual de Usuario 25: The last step in EPF Role Creation Wizard, alert of role created	Pág. 273.
Manual de Usuario 26: Press this button to create a task in your project	Pág. 273.
Manual de Usuario 27: Press this button to add EPF info to your task ..	Pág. 274.
Manual de Usuario 28: The button that launches the Role Creation Wizard in the button "New role"	Pág. 275.
Manual de Usuario 29: The button that launches the Role Creation Wizard in the tab "EPF Roles"	Pág. 275.
Manual de Usuario 30: The first step in Workproduct Creation Wizard..	Pág. 275.
Manual de Usuario 31: The second step in Workproduct Creation Wizard. Initial aspect.	Pág. 276.
Manual de Usuario 32: The second step in Workproduct Creation Wizard, with roles and tasks selected	Pág. 278.

Manual de Usuario 33: The third step in Workproduct Creation Wizard, Configuring users for roles selected.....	Pág. 279.
Manual de Usuario 34: The last step in Workproduct Creation Wizard, alert of workproduct created	Pág. 279.
Manual de Usuario 35: The button that launches the wizard to create a new template	Pág. 280.
Manual de Usuario 36: Two steps to edit a role	Pág. 281.
Manual de Usuario 37: Press this button to edit your task EPF data	Pág. 282.
Manual de Usuario 38: You can edit these options of your workproduct	Pág. 283.
Manual de Usuario 39: Delete confirmation page	Pág. 285.
Manual de Usuario 40: You cannot delete a role if it has dependencies on it	Pág. 285.
Manual de Usuario 41: In this box you'll find the EPF info of your projects	Pág. 287.
Manual de Usuario 42: Tabs inside the project info page.....	Pág. 287.
Manual de Usuario 43: Workproduct tab contents	Pág. 288.
Manual de Usuario 44: Role information page	Pág. 289.
Manual de Usuario 45: Task information page	Pág. 290.
Manual de Usuario 46: Workproduct information page.....	Pág. 291.
Manual de Usuario 47: Template information page.....	Pág. 292.
Manual de Usuario 48: Tabs in the user info page: General.	Pág. 293.
Manual de Usuario 49: Tabs in the user info page: EPF Specific Roles.	Pág. 293.
Manual de Usuario 50: Tabs in the user info page: EPF Tasks.	Pág. 294.
Manual de Usuario 51: Workproduct: You have no permission to edit this workproduct status.....	Pág. 295.
Manual de Usuario 52: Modifying your workproduct status	Pág. 295.
Manual de Usuario 53: Reports link to access to the reports list	Pág. 296.
Manual de Usuario 54: New FVE Report.....	Pág. 296.
Manual de Usuario 55: New FVE Project Manager report	Pág. 297.
Manual de Usuario 56: Subversion Tab Interface in dotproject.....	Pág. 298.

Manual de Usuario 57: Subversion project information page.....	Pág. 299.
Manual de Usuario 58: Subversion FileManager Interface.....	Pág. 300.
Manual de Usuario 59: Initial contents of a subversion project (Showing FileManager Interface)	Pág. 301.
Manual de Usuario 60: Commit files Subversion Interface	Pág. 302.

1. Introducción

FVE Project Manager es el nombre de la aplicación que se ha desarrollado en este proyecto. El nombre resulta de la intención de reunir en una sola palabra a sus autores, tomando las iniciales de sus apellidos: Fábregas, Vilela y Estirado. Este proyecto tiene como objetivo proporcionar una solución de código abierto que integre un gestor de proyectos con un sistema de control de versiones, y que proporcione flexibilidad en la definición e instanciación de procesos.

La aplicación permite la instanciación de un proyecto cuya estructura (roles, productos de trabajo, tareas, etc.) está definida en Eclipse Process Framework Composer para diferentes metodologías de desarrollo. Incluye además el soporte de un sistema de control de versiones (Subversion) integrado en un gestor de proyectos, y la posibilidad de configuración y de trabajo diario en un proyecto con las nuevas entidades generadas desde un gestor de proyectos (dotProject) que permita adaptarse a los cambios prácticamente inevitables en cualquier planificación. A lo largo de la memoria se realiza un profundo análisis de estas herramientas implicadas, incluyendo información sobre todas las opciones estudiadas y los motivos de elección entre unas y otras, hasta adentrarse en la propia estructura del proyecto, detallando en qué consiste la nueva funcionalidad y cómo se ha llevado a cabo la integración entre todas las herramientas (se detalla en el siguiente apartado con más detalle la estructura de la memoria).

El desarrollo de la aplicación se hizo de acuerdo a un planificación previa. Se dedicó una primera fase del proyecto al estudio de aplicaciones de gestión y de control de versiones, seleccionando las alternativas Open Source con mayor número de usuarios y delimitando los requisitos que debía cumplir el proyecto a partir de la funcionalidad que ya aportaban dichas alternativas. El gestor de proyectos elegido fue dotProject, con más de 60.000 descargas en su última versión [SFPD 08] y un número de usuarios aún más alto. Para la gestión de versiones se eligió Subversion (SVN), herramienta de referencia en el desarrollo Open Source. Tras la primera fase de estudio, se pasó a la fase de desarrollo realizando una serie de entregas centradas en la incorporación de mejoras incrementales en el gestor de proyectos. Esto fue posible gracias a la tendencia en este tipo de desarrollos a favorecer la creación de módulos.

La decisión de trabajar con soluciones Open Source ya existentes, aportando nuevas funcionalidades, se basó en las ventajas asociadas a este método de trabajo: Por una parte se posibilitaba de esta forma que el trabajo pudiera ser continuado por una comunidad de desarrolladores fuera del espacio de la asignatura, asegurando una plataforma central para mostrar y publicitar el trabajo. Por otra, se permitía definir objetivos más ambiciosos debido a construir el proyecto a partir de un código disponible

que cumplía las opciones más básicas del proyecto: Esto exigía un sobreesfuerzo inicial de análisis y adaptación, pero permitía después un avance más rápido y la posibilidad de profundizar más en la gestión de procesos y control de versiones. Por todas estas razones se considera justificado aprovechar la posibilidad de la asignatura Sistemas Informáticos para asumir desde el primer momento un enfoque claro de apoyo a proyectos Open Source.

Aparte de la herramienta FVE Project Manager, se han generado ejemplos de uso de la herramienta y un manual detallado con todas las opciones de la aplicación en inglés. La decisión de elaborar todo el software en inglés, incluyendo el manual de instalación, el manual de usuario y la metodología de ejemplo entregadas, refrenda el compromiso de llegar a un gran número de usuarios y facilitar la implantación de FVE Project Manager para que el trabajo desarrollado no se limite al ámbito académico.

1.1 – Estructura de la memoria

La memoria que se entrega en el presente documento está estructurada en 11 secciones y 7 anexos, incluyendo en primer lugar y antes del índice los contenidos definidos como obligatorios en la normativa de Sistemas Informáticos: Portada normalizada, Autorización de uso, Resumen y Palabras Clave (en castellano e inglés). Tras esta información se analiza el uso de las herramientas implicadas en el proyecto, justificando su uso y evaluando sus posibilidades, llevando a los Anexos B, C y D los documentos realizados más específicos sobre estas herramientas (Trabajos relacionados con gestores de proyectos, EPF y Subversion). La siguiente sección se refiere al Análisis del proyecto, y contiene la información asociado a la captura de requisitos y casos de uso en la aplicación. La siguiente sección, Desarrollo, profundiza en los aspectos estructurales e internos de la aplicación, resumiendo el trabajo realizado a distintos niveles. A continuación se presenta la sección de Organización del proyecto, que resume el ciclo de vida del proyecto y las entregas realizadas durante la asignatura, y especifica las labores llevadas a cabo o dirigidas por cada integrante del grupo. La siguiente sección, Ejemplo de Uso, muestra las ventajas del uso de la aplicación resultante en un proyecto real, aportando diferentes usos del programa. A continuación se incluye la sección Conclusiones y trabajo futuro, que analiza los resultados del proyecto y evalúa de forma crítica aspectos que podrían ser mejorados en el futuro. La última sección se refiere a la bibliografía empleada en el proyecto, que es referenciada desde cada punto de la memoria que está basado o recoge influencias de una obra o publicación externa.

Los anexos añadidos, aparte de los ya citados de Trabajos relacionados con herramientas, contienen una guía de estilo definida para trabajar con la memoria, el análisis de los prototipos previos realizados y las

ideas que éstos contenían y que en algunos casos sirvieron para definir la aplicación final, y un manual de instalación y de usuario en inglés que explican el funcionamiento interno de cada opción de la aplicación, incluyendo capturas de pantalla, solución de errores frecuentes y explicaciones detalladas sobre el uso de FVE Project Manager. Los dos últimos anexos han sido realizados en inglés, junto a la metodología desarrollada como ejemplo en EPF Composer y explicada en la sección 2.2.6, tras petición por escrito al Vicedecano de Ordenación Académica, para poder generar material liberado a la comunidad internacional de usuarios de dotProject y facilitar el uso futuro de la herramienta.

2. Estado del arte

Se conoce como estado del arte a la enumeración de todos los progresos llevados a cabo dentro de un campo, en este caso la gestión de proyectos, y que tiene como fin justificar las novedades introducidas dentro de ese mismo campo.

Con esta premisa, nuestro proyecto sirve como elemento de unión entre numerosas herramientas con objetivos muy diferenciados, y por tanto es necesario integrar sus funcionalidades en una nueva. Entre todas las herramientas que pueden desempeñar las funciones establecidas en los requisitos del proyecto se realiza una evaluación, para justificar las herramientas finalmente utilizadas.

2.1 Gestores de proyectos

2.1.1 Función

La gestión de un proyecto es una tarea compleja que involucra una serie de aspectos multidisciplinarios y que lleva asociada una carga de trabajo importante. Teniendo en cuenta que la gestión de un proyecto está determinada por el campo en el que se desempeñe el proyecto en primera instancia, y por las particularidades del proyecto después, el objetivo de los directores de proyectos ha ido siempre encaminado a automatizar las tareas que implican la gestión de un proyecto. Esta uniformidad es más complicada a medida que se abre el campo de los proyectos tratados, por lo que las primeras medidas siempre tratan de marcar objetivos comunes en el entorno de una empresa o un departamento concreto.

La aparición de soluciones informáticas que trataran de facilitar la gestión de un proyecto, teniendo en cuenta las complicaciones ya señaladas en el anterior párrafo, ha sido realmente muy abundantes y se han desarrollado aplicaciones desde hace mucho tiempo. No obstante, a medida que se extendió el uso de Internet las aplicaciones dedicadas a la gestión de proyectos sufrieron una evolución muy importante y surgieron un gran número de soluciones webs que trataban de aportar soluciones genéricas y un entorno multiusuario alejado de herramientas destinadas únicamente al director de proyecto. Los nuevos entornos colaborativos (basados en conceptos recientes asociados a la web 2.0), las interfaces más claras y relacionadas accesibles desde cualquier ordenador con acceso web, y el impacto de multiplicar el número de usuarios de cada aplicación debido al auge del desarrollo Open Source y de los usuarios interesados en estas soluciones han sido los factores clave de que este sea uno de los campos que más usuarios e inversiones aglutina.

Un gestor de proyectos que se considere genérico y salvando las excepciones de cada entorno y cada finalidad concreta, suele disponer de:

- Gestión de tareas, incluyendo información sobre recursos dedicados, usuarios y fechas.
- Gestión de usuarios, permitiendo a cada uno estar al tanto de sus tareas y proyectos, y comunicación con otros.
- Información sobre la ejecución del proyecto, desde numerosos puntos de vista: Diagramas técnicos, reportes, etc.

2.1.2 Justificación de aplicación en el proyecto

Uno de los objetivos principales de nuestro proyecto es aportar una solución que gestione procesos de desarrollos software, de modo que se profundice en un tipo de proyecto (informático, con un perfil más técnico), alejándose a primera vista del enfoque actual de los gestores de proporcionar soluciones genéricas. Sin embargo, lo más adecuado para un usuario habitual de gestores de proyecto es que este esfuerzo en particularizar el tipo de proyecto no implicara el uso de una nueva herramienta, sino redefinir una herramienta existente que ya utilizara y cuyo uso estuviera ya implantado. Los riesgos de un nuevo gestor de proyectos son numerosos y se pueden reseñar aquí: Existen muchas funcionalidades comunes que es necesario volver a implementar, es necesario actualizar el software con frecuencia para estar al tanto de todas las novedades que se aporten para todo tipo de proyectos (aunque no fueran los que en principio se querían tratar), y es necesario conseguir una cuota de mercado para poder tener apoyo en forma de usuarios que es prácticamente imposible con la de opciones que hay en la web. De este modo, si se realiza un buen gestor centrado en un aspecto muy concreto, el siguiente paso en los actuales gestores de proyectos sería que algún colaborador creara un plugin para integrar esa herramienta en el gestor general.

La alternativa de integrar funcionalidad en un gestor de proyectos Open Source existente también tiene ciertas desventajas que es necesario tener en cuenta y evaluar en los gestores disponibles: Trabajar con un gestor de proyectos existente supone un esfuerzo adicional y previo a la implementación debido al necesario estudio de la arquitectura del sistema, su soporte de módulos, su estilo de programación, y las limitaciones que impone en lenguaje y estructura. Conseguir que los nuevos contenidos a introducir sean visibles y compatibles con las estructuras generales de la aplicación a completar requiere de un esfuerzo por crear contenidos compatibles, intentando evitar la redundancia de contenidos.

Analizadas las ventajas e inconvenientes, se decidió realizar un estudio de los gestores open source más conocidos y completos para que si

uno de ellos cumplía los requisitos mínimos de modularidad y complemento de los objetivos del proyecto realizar nuestro proyecto sobre él. Así, además, se cumplía el objetivo de realizar una aportación destacable en un proyecto Open Source desde el ámbito académico y se dotaba de posibilidades al proyecto para que fuera utilizado por un mayor número de usuarios, respondiendo a una necesidad real y no limitando su ciclo de vida únicamente a la duración de la asignatura de Sistema Informáticos.

2.1.3 Estudio de gestores

En la primera fase de investigación se descubrió una gran diversidad de gestores de proyectos, y por ello se elaboró un listado con los nombres de estas herramientas obteniendo más de veinte aplicaciones. No obstante, se seleccionaron tras un primer examen los gestores de proyectos que debido a su naturaleza Open Source, su número de usuarios y sus características distintivas mejor se adaptaban al objetivo del proyecto. Las herramientas seleccionadas y sobre las se realizó un examen más detallado, aportando el estudio incluido en el anexo B.1, fueron: dotProject, EgroupWare, PhpCollab, PhpProjekt, Open Workbench, Web2project, ActiveCollab y Trac, descartando ya en la primera fase otras soluciones como PhpGroupWare, drProject, BaseCamp, FitBak, Net-Office, Tutos y WebCollab entre otras.

2.1.4 Elección de gestor: dotProject

La importancia de la elección de un gestor de proyecto ya ha sido debidamente destacada en los apartados previos y por tanto esta decisión fue la que más esfuerzo requirió en la etapa de planificación del proyecto. La decisión de adoptar dotProject, apoyada en los argumentos que ya se mencionan en la tabla, se debió a los siguientes factores:

- Gestor de proyecto más utilizado: No es un factor que por sí solo tenga influencia para realizar el proyecto sobre esta aplicación, ya que podría por propiedades internas ser el más inadecuado, pero le coloca en una posición preferente entre los estudiados ya que utilizar dotProject supone llegar al mayor número de usuarios. Además, ser el más utilizado suele implicar (y en este caso se ha comprobado que es así) que ser el más utilizado supone ser el que más documentación existente tiene, el que mayor número de módulos creados por usuarios externos tiene, el que está traducido a más idiomas con un soporte multiplenguaje adecuado, etc.
- Sistema de soporte de módulos: Es claramente el mejor en este apartado, aunque eso no signifique que el soporte para módulos sea perfecto. Presenta unos convenios que pueden adoptarse para que la instalación de módulos sea sencilla [dPDocumentation1 08], se pueda incluir toda la información necesaria para el módulo incluyendo clases, creación de bases de datos, etc., y al haber sido realizados algunos módulos ya en esta

aplicación los riesgos de encontrarse problemas al realizar módulos disminuyen.

Para consultar información más específica sobre dotProject, sobre su estructura y funcionamiento, se remite al lector al anexo B.2.

2.2 EPF: Gestión de procesos de desarrollo

2.2.1. Eclipse Process Framework

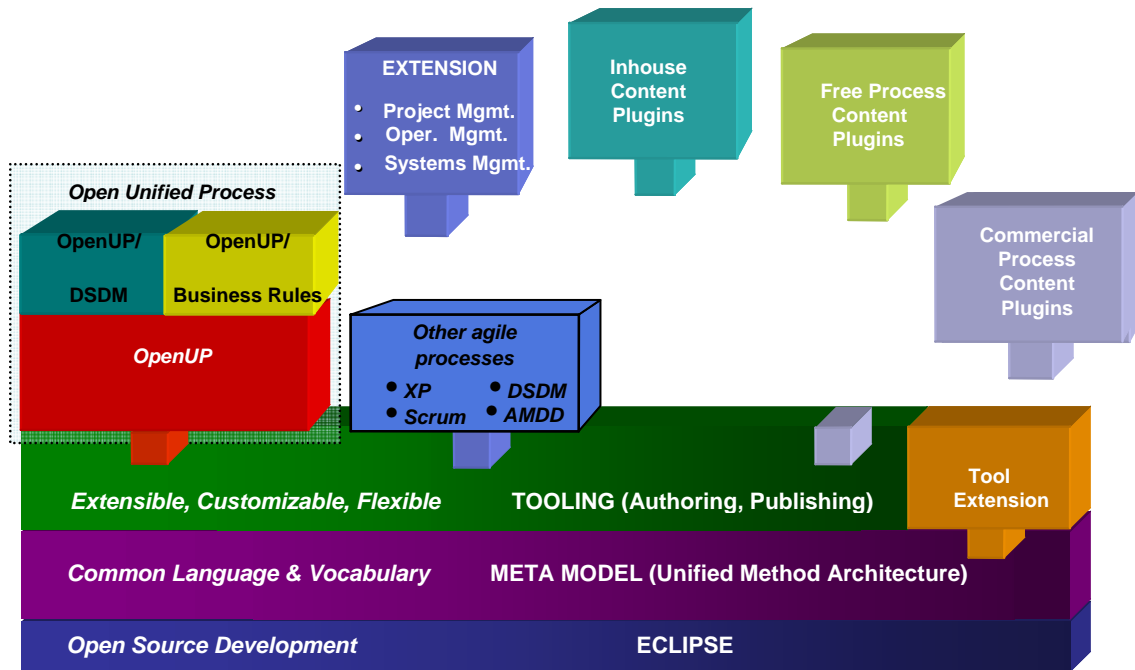
EPF Logo

El proyecto Eclipse fue originalmente creado por IBM en Noviembre de 2001 y fue apoyado por otras compañías. La organización que debería gestionar este proyecto general se denominó Eclipse Foundation, y se constituyó definiendo unas áreas de trabajo en las que se enmarcarían sus proyectos concretos [EclipseFoundation1 06]. Eclipse Process Framework es un proyecto que comienza en enero de 2006 y se enmarca como parte de los proyectos de Tecnología de la Eclipse Foundation, tratando de lograr diversos objetivos:



- Servir como organización para agrupar el trabajo Open Source dedicado al desarrollo de procesos software.
- Proveer herramientas, un modelo de datos unificado y contenidos que puedan ser usados por la fundación y cubran las necesidades de los proyectos de las tecnologías de la información. Como herramienta se aporta un framework extensible para trabajar con ingeniería de procesos como autor, manejo de librerías de procesos, configuración y métodos de publicación de los procesos. Como contenido se aportan diferentes procesos detallados destacando la creación de un proceso de desarrollo nuevo, OpenUP, que aglutina los conceptos del Proceso Unificado de Desarrollo con las pautas del Manifiesto Ágil que busca la eficiencia y el mejor ajuste de los procesos.

Eclipse Process Framework Composer (EPF Composer) es el software que cumple con la propuesta de proveer una herramienta extensible de creación y configuración de procesos de desarrollo. Este software se integra dentro del interfaz de Eclipse, aunque no está restringido su uso para uno u otro lenguaje ni tiene dependencia su contenido con el entorno de desarrollo que se use en el proceso de implementación del proyecto, sino que su función es paralela al desarrollo propio del proyecto. Su arquitectura en capas se muestra en este esquema:



([EclipseFoundation2 05]. *What is the Eclipse Process Framework - Eclipse Foundation Presentation.*)

2.2.2. Comparación con Rational Method Composer: Similitudes y Carencias

EPF Composer se considera la alternativa Open Source del producto comercial IBM Rational Method Composer (RMC), encargado de la autoría, configuración, visión y publicación de procesos software. IBM además proporciona un catálogo de procesos predefinidos (cuyo ejemplar más conocido y con más aceptación es Rational Unified Process) que pueden ser adaptados a proyectos concretos. Incluyen asimismo bloques que especifican las mejores prácticas de desarrollo para disciplinas, tecnologías o desarrollos específicos que denominan Capability Patterns.

EPF Composer se basa en el estándar SPEM (Software Process Engineering Metamodel) como metamodelo de procesos software. SPEM es una propuesta del grupo OMG (Object Management Group) que establece las pautas y estándares en el campo de los procesos software, y está actualmente en su versión 2.0. El metamodelo profundiza en 5 núcleos conceptuales: Method content, delivery processes, capability patterns, categories y plug-ins. No obstante estos conceptos son opcionales y se pueden crear procesos sin que sigan exactamente esta estructura. La solución RMC también presenta compatibilidad con SPEM, estando ambas herramientas desarrollando y completando sus funcionalidades para soportar todas las opciones definidas en la versión de SPEM 2.0 [OMG1 07].

Una de las características relevantes de RMC son los plugins que aporta a sus procesos predefinidos, como los módulos "RUP for SOA" (que define procesos del modelo unificado de desarrollo aplicables a arquitecturas orientadas a servicios), RUP SE (aplicable a ingeniería de sistemas), etc. Es en este nivel de detalle donde comienzan las diferencias con el nivel actual de EPF Composer, aplicación que completa con un gran nivel de detalle el proceso OpenUp pero que no presenta plugins específicos dependientes de la categoría de un proyecto. Y debido a esta diferente estructura se considera que los procesos desarrollados en EPF Composer requieren de un tiempo de creación muy elevado, reutilizando en menor forma los contenidos que la solución comercial de IBM.

Sin embargo, la carencia principal que presenta Eclipse Process Framework respecto a Rational Method Composer es la falta de integración con otras herramientas. Así, mientras la solución comercial RMC puede conectarse con otras herramientas de IBM Rational, como Rational Portfolio Manager o Rational Process Workbench, las opciones de EPF Composer solo pueden exportarse, de manera circunstancial, a Microsoft Project [Wikipedia108]. Estas características posibilitan una mayor seguridad para un usuario que puede confiar en una solución sabiendo que le aporta posibilidades en diversos campos, mientras que EPF se centra más en únicamente documentar el proceso, y las posibilidades de conexión con otras herramientas son muy reducidas. En la gestión de un proyecto la definición y/o configuración de una metodología a seguir es tan solo una fase de los proyectos por lo que el usuario debe contar con otras herramientas y multiplicar el tiempo empleado en definir un proceso repitiendo contenidos en diversas herramientas, perdiendo en parte la exactitud de lo definido en EPF Composer.

2.2.3. Elementos de EPF Composer

EPF Composer define una serie de elementos que son los que forman un proceso de desarrollo, la entidad superior de la aplicación. De acuerdo a los principios de Ingeniería de Procesos Software y basado en las definiciones del Proceso Unificado de Desarrollo de IBM se definen una serie de elementos que especificamos en esta sección, y que serán necesarios en su gran mayoría en la aplicación realizada [EPF108]:

- Patrones genéricos de procesos (Capability Pattern): Este tipo de construcción es un tipo especial de procesos que describen un conjunto de actividades reutilizables y aplicables a un área común de un proceso de desarrollo.
- Proceso de desarrollo (Delivery Process): Un proceso de desarrollo es un proceso que cubre el ciclo de vida completo de un proyecto, con fases, actividades e iteraciones predefinidas.

- Rol: Se trata de una serie de habilidades, competencias y responsabilidades que forman una entidad, que puede adoptar un usuario o varios y que especifica quién debe trabajar en el desarrollo de tareas y productos de trabajo.
- Producto de Trabajo: Es el término general adoptado para los artefactos que se generan, modifican o utilizan en una tarea.
- Guía: Es cualquier tipo de ayuda que sirve para conocer cómo debe llevarse a cabo una tarea o producto de trabajo, como ejemplos, plantillas o listas de cuestiones a comprobar durante el desarrollo de una acción.
- Actividad: Constituyen los bloques de los procesos de desarrollo y agrupan a los demás elementos (incluso a nuevas actividades).
- Iteración: Un grupo de actividades que pueden repetirse en más de una ocasión.
- Fase: Es un tipo de actividad que representa un periodo significativo en el proyecto.
- Tarea: Se trata de una unidad de trabajo, generalmente con una duración de algunas horas o pocos días, y es el elemento principal de un proyecto.
- Definiciones de términos: Se trata de definiciones clave del proyecto que detallan su ámbito y significado. Se incluyen en un glosario para evitar ambigüedades.

2.2.4. Uso de EPF Composer

Debido a la importancia de esta herramienta en el proyecto se ha hecho un uso completo de la herramienta y se ha profundizado en todas las fases de la construcción de un proceso de desarrollo. La documentación asociada sobre su uso y metodologías definidas se incluye en el Anexo C.

2.2.5. Justificación de aplicación en el proyecto

Sustentado en los puntos anteriores, se definen los puntos principales por los que esta herramienta se aplica en el proyecto:

- Se trata de una herramienta que forma parte de Eclipse Foundation, por lo que se espera el desarrollo y auge de la herramienta. Además, está probado el soporte de la comunidad como proyecto Open Source (el más importante que entra en todos los detalles de un proceso de desarrollo y que compite con Rational Method Composer) y de empresas como IBM.
- La definición de los procesos de desarrollo es muy completa, y cubre los objetivos que pueden esperarse de un gestor de procesos de desarrollo, al margen de que el diseño de la herramienta o las opciones de exportación sean menores que en otros gestores de procesos.

- Se cubren numerosas necesidades de herramientas que no existen en el mercado: Como ya se ha explicado, el punto débil de EPF Composer se considera que es la falta de herramientas que puedan aprovechar los procesos de desarrollo creados en ésta, por lo que aportar una herramienta capaz de importar esta información a un gestor de proyectos web supone aportar una solución no existente y favorecer el impulso de EPF. Además, no existen gestores de proyectos web con información tan detallada de los procesos de desarrollo, al margen de que el gestor de procesos sea o no EPF Composer, por lo que el usuario que quiera trabajar con un gestor de proyectos y estas ideas obtiene una solución que tampoco existía antes de este proyecto.

2.2.6. Definición en detalle de una metodología (aplicable a proyectos de Ingeniería del Software a partir de la experiencia de los integrantes del grupo) usando EPF Composer.

Para poder desarrollar más adelante un ciclo completo del proyecto desde la instanciación de un proceso de desarrollo hasta el trabajo diario con un proyecto concreto, se creó un proceso de desarrollo completamente nuevo que trata de resumir la experiencia de los integrantes del grupo en la asignatura Ingeniería del Software creando una metodología que recoge la planificación que se siguió en la asignatura. Esta experiencia está apoyada en los entregables generados en 2006/2007 en el proyecto SimTraffic (simulador de tráfico vehicular) [GrupoSimTraffic 07], pero se abstrae del objetivo concreto de este proyecto para ser aplicable a cualquier otro proyecto que se diera en esa asignatura.

Se muestran los resultados más relevantes de la metodología desarrollada, aunque debido a la extensión de la misma no se incluye todos los diagramas generados ni las descripciones detalladas de todos los elementos. En cualquier caso, la metodología completa se trata de uno de los entregables del proyecto. La metodología ha sido generada en inglés para presentarla como ejemplo completo del proyecto, que está realizado en inglés, y poder llegar a un mayor número de usuarios.

2.2.6.1 - Definición de iteraciones

El proceso de desarrollo ha sido agrupado en 11 iteraciones:

Presentation Name	Index	Predecessors	Model Info	Type	Planned
IterativeSoftwareEngineerUCMProcess	0			Delivery Process	<input checked="" type="checkbox"/>
Propose a idea for a project	1			Iteration	<input checked="" type="checkbox"/>
Define requirements	6	1		Iteration	<input checked="" type="checkbox"/>
Investigate technologies	13			Iteration	<input checked="" type="checkbox"/>
Investigate Domain	19	7,8		Iteration	<input checked="" type="checkbox"/>
Make a design	24	6,13,19		Iteration	<input checked="" type="checkbox"/>
Organize work	31	6,13,19		Iteration	<input checked="" type="checkbox"/>
Implementation	39	24,31		Iteration	<input checked="" type="checkbox"/>
Organize test for public	46	39		Iteration	<input checked="" type="checkbox"/>
Debug project	52	39		Iteration	<input checked="" type="checkbox"/>
Revise software documentation	59	24,31		Iteration	<input checked="" type="checkbox"/>
Publish project	64	52,59,46		Iteration	<input checked="" type="checkbox"/>

Listado de iteraciones.

Las iteraciones no son estrictamente cronológicas en el orden que figuran, por lo que se han incluido dependencias lógicas entre ellas que muestra el siguiente diagrama.

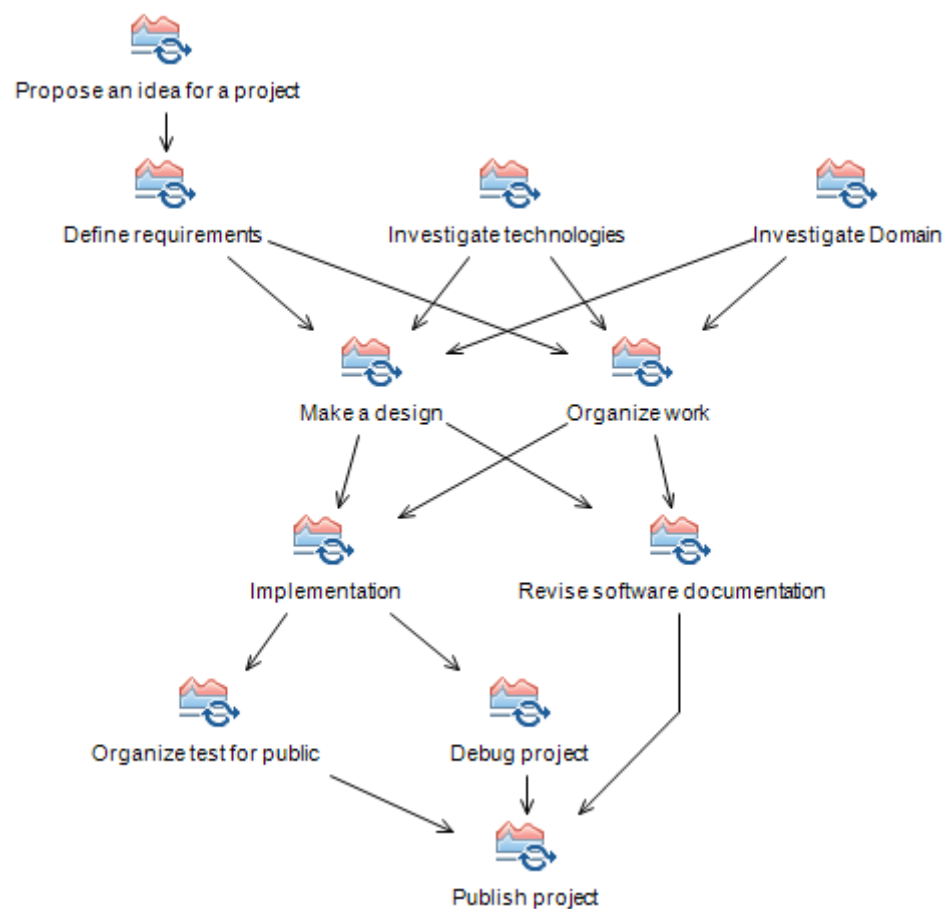


Diagrama de dependencias entre iteraciones.

Puede consultarse el contenido concreto para cada iteración obteniendo una descripción detallada de sus finalidades.

2.2.6.2 - Definición de tareas

En cada iteración se han incluido las tareas que se realizaron durante el transcurso de la asignatura Ingeniería del software englobadas en las iteraciones anteriores, con un total de 59 tareas.

Se listan 3 ejemplos, con diagramas de actividad generados en los que se muestran para cada iteración las tareas implicadas, teniendo en cuenta las dependencias introducidas.

- Propose an idea:

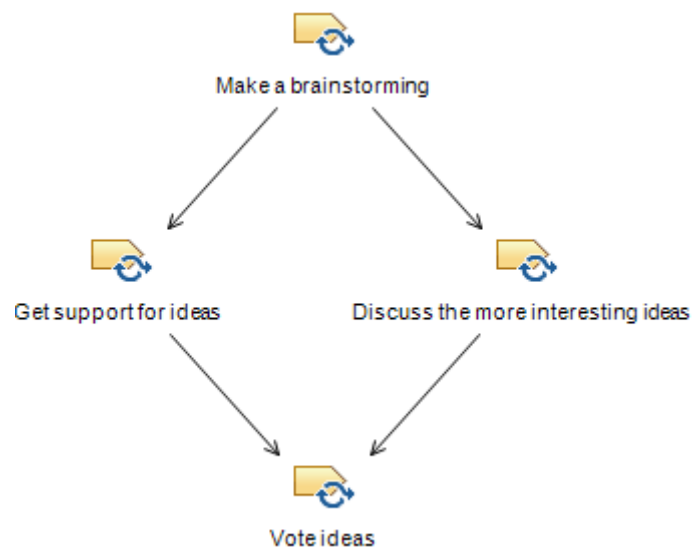


Diagrama de actividad 1: Propose an idea.

- Investigate technologies:

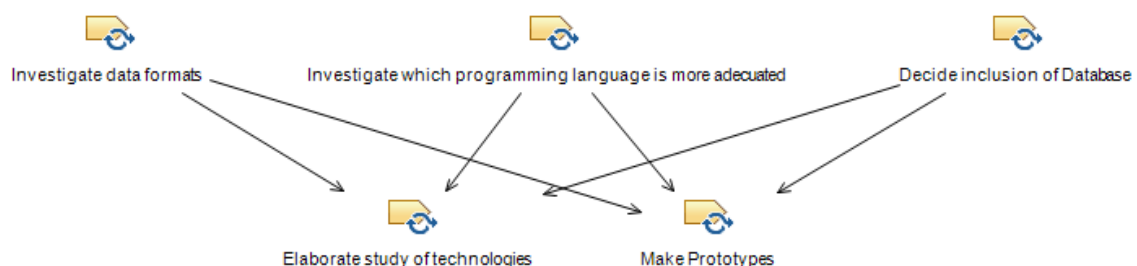


Diagrama de actividad 2: Investigate technologies.

- Publish Project

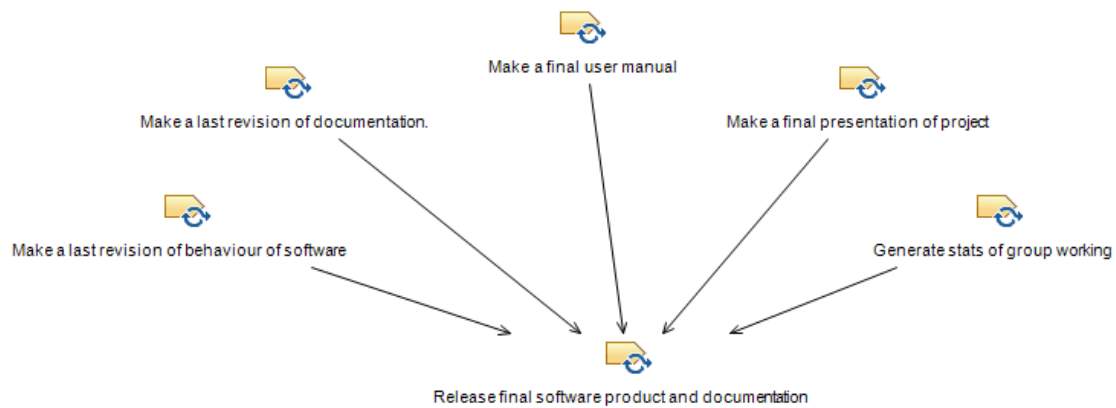


Diagrama de actividad 3: Publish Project.

2.2.6.3 - Definición de roles

Se han definido 14 roles que delimitan las obligaciones que deben asumir los usuarios cuando quieran participar en tareas o productos de trabajo. Estos roles no implican que solo haya una persona que los desempeñe, ni que un usuario sólo pueda asumir un rol. Además, una vez definidos los roles, se han asignados usuarios principales y adicionales a las tareas.

Los roles generados son los siguientes:





















Analyst	Role Descriptor
Documentor	Role Descriptor
DomainInvestigator	Role Descriptor
InspectorCode	Role Descriptor
InspectorDocumentation	Role Descriptor
Leader Of Subgroup	Role Descriptor
Non-Domain Investigator	Role Descriptor
Participant In Presentations	Role Descriptor
Programmer	Role Descriptor
Project Manager	Role Descriptor
QAInspector	Role Descriptor
Reunion Secretary	Role Descriptor
SpokesMan	Role Descriptor
Tester	Role Descriptor

Listado de roles.

Las tareas pasan a tener un rol primario y uno o más roles adicionales.

2.2.6.4 - Definición de productos de trabajo

Se han elaborado 20 entregables a lo largo de la asignatura, que pueden estar sujetos a actualizaciones y nuevas versiones porque en su mayoría participan en numerosas iteraciones (por ejemplo, el documento de riesgos debe actualizarse con frecuencia en cada iteración del sistema). El listado de workproducts se muestra a continuación:

Presentation Name	Model Info	Type
 Technologies Report	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Non Functional Requirements	Mandatory Input,Output	Artifact Descriptor
 Presentations	Mandatory Input,Optional Input,Output	Artifact Descriptor
 PlanProject	Mandatory Input,Output	Artifact Descriptor
 Quality Assurance Report	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Risk List	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Configuration Management Report	Mandatory Input,Output	Artifact Descriptor
 Formal Technique Revisions	Mandatory Input,Output	Artifact Descriptor
 OnePage Idea	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Concrete details of domain	Mandatory Input,Optional Input,Output	Artifact Descriptor
 CRC Cards	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Design Explanation	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Functional Requirements	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Software Requirements Specification	Mandatory Input,Output	Artifact Descriptor
 UML Design	Mandatory Input,Optional Input,Output	Artifact Descriptor
 UseCases Document	Mandatory Input,Optional Input,Output	Artifact Descriptor
 User Manual	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Stories	Mandatory Input,Optional Input,Output	Artifact Descriptor
 Estimation	Mandatory Input,Output	Artifact Descriptor
 Document of application of design patterns	Mandatory Input,Optional Input,Output	Artifact Descriptor

Listado de productos de trabajo.

En este momento ya pueden añadirse dependencias entre roles y productos de trabajo, pudiendo ya mostrar cómo las tareas (primera columna) y los productos de trabajo (segunda columna) son asignados a un rol concreto, Analyst:

Role: Analyst



The analyst in this project has high level functions in software development as discuss and define design, case-uses, etc in the application. The main difference between an analys and a programmer is the higher responsibility in design and less in code implementation matters.

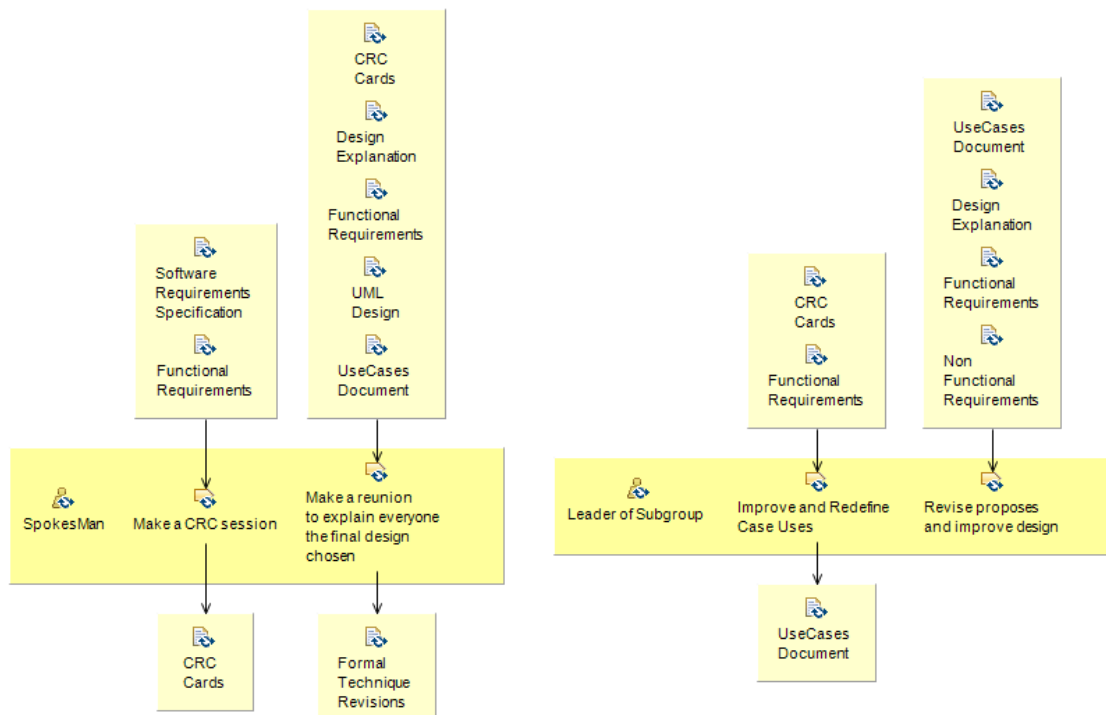
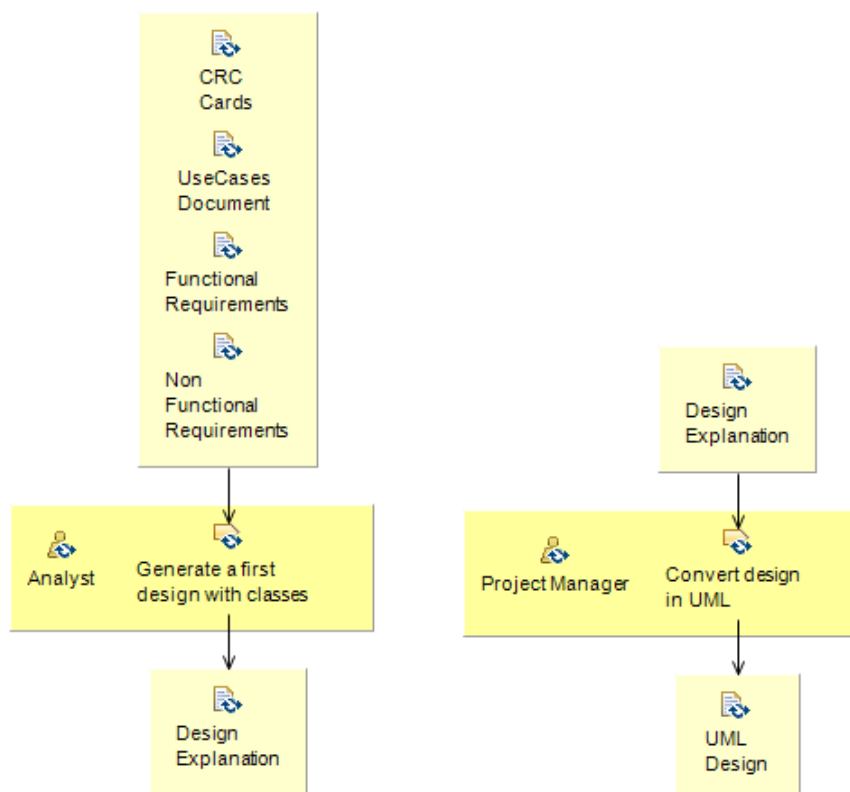
Expand All Sections

Collapse All Sections

Relationships	
Performs: <ul style="list-style-type: none"> • Create Case Uses • Propose Functional Requirements 	Responsible for: <ul style="list-style-type: none"> • Design Explanation • Document of application of design patterns • Functional Requirements • Software Requirements Specification • UML Design • UseCases Document
Additionally Performs: <ul style="list-style-type: none"> • Contrast opinions with other members • Imagine situations in which the application could be useful • Implements automatic tests • Update the risk list in order to obtain conclusions. • Writing histories 	Modifies: <ul style="list-style-type: none"> • Functional Requirements • Risk List • Software Requirements Specification • UseCases Document

Role Analyst en detalle.

Además, se han indicado las dependencias entre las tareas y los roles con cada workproduct generado, obteniendo para tres iteraciones de ejemplo los siguientes diagramas de actividad detallados que muestran los workproducts asociados a las tareas.

- Make a Design:**Diagrama de actividad detallado 1: Make a Design.****Diagrama de actividad detallado 2: Make a Design.**

- Investigate technologies:

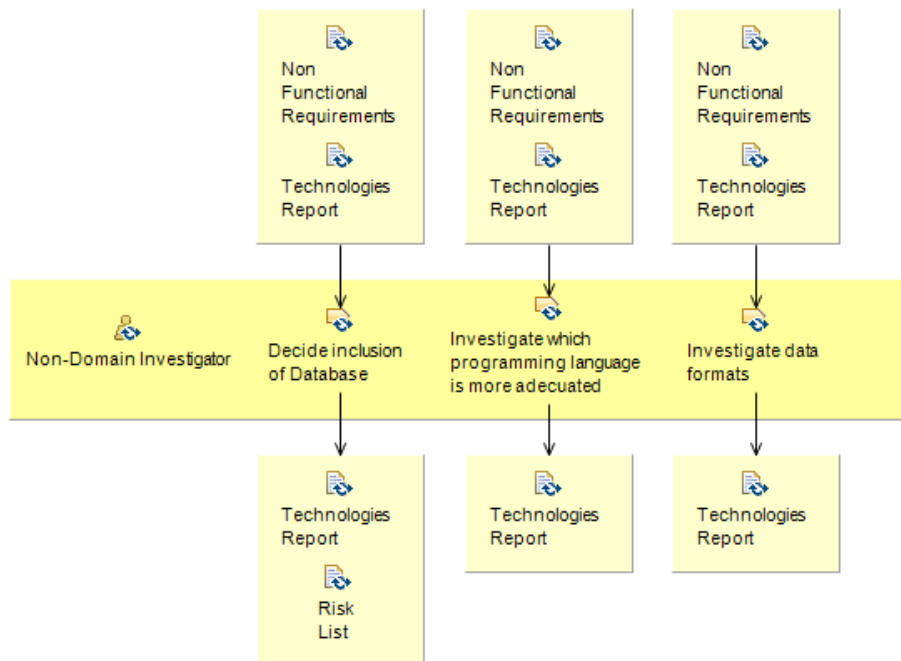


Diagrama de actividad detallado 1: Investigate technologies.

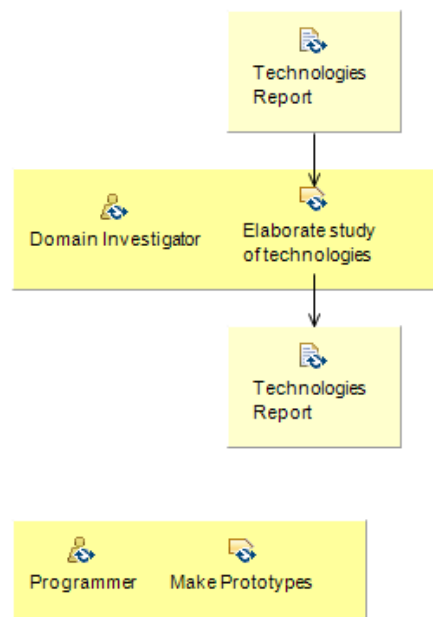


Diagrama de actividad detallado 2: Investigate technologies.

2.2.6.5 - Exportación XML

Se finalizó el proceso generando un fichero xml que se empleará en el gestor de proyectos al importar el proceso de desarrollo creado. Este fichero contiene toda la información generada, y tiene un tamaño de 139 Kb.

2.3 Repositorio de control de versiones

Un sistema de control de versiones se encarga de guardar revisiones de los ficheros en un repositorio (almacén) para facilitar el trabajo en grupo y la recuperación de versiones anteriores. Ésta constituye una buena práctica que está ampliamente extendida en el sector informático y que está en continua mejora [Fowler 00]. Existen en el mercado numerosas opciones de software que realizan esta tarea, aunque son dos herramientas Open Source, CVS y Subversion (SVN), las más utilizadas. Se realiza un estudio de estos sistemas y de herramientas que facilitan su gestión para encontrar un método de integrar esta funcionalidad en un gestor de proyectos.

2.3.1 Justificación de aplicación en el proyecto

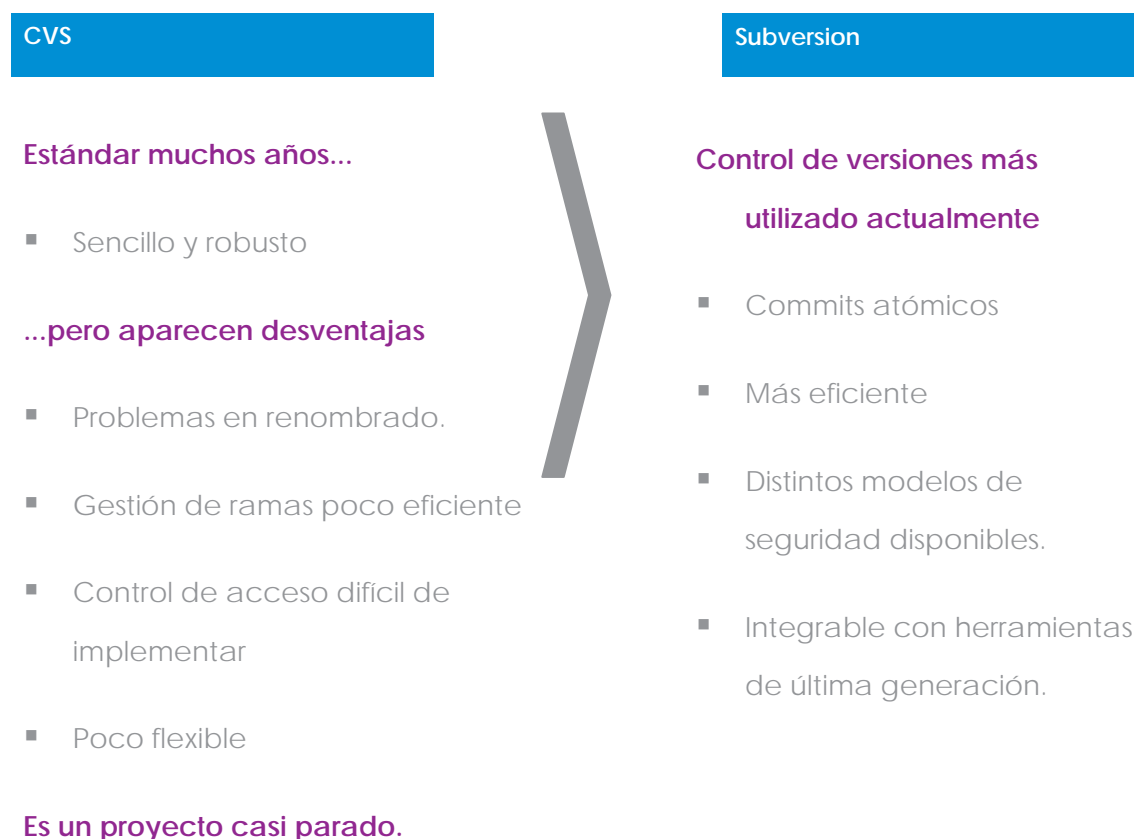
Uno de los requisitos marcados en el proyecto fue la inclusión de un sistema de control de versiones en un gestor de proyectos. Este objetivo no tenía únicamente su base en las innegables ventajas de trabajar con un sistema de control de versiones, ya que éste es un sistema empleado en prácticamente todos los proyectos informáticos en la actualidad, sino en integrar esta funcionalidad en un gestor de proyectos. No existen gestores de proyectos web Open Source, como dotProject o phpCollab, que incluyan módulos de control de versiones, por lo que incluir este servicio supone una herramienta única y que responde a una necesidad como prueban las peticiones en los foros y listas de correo de las respectivas aplicaciones. Este sistema debería servir tanto para tratar tanto el código fuente como la documentación del proyecto, aunque debe hacerse un esfuerzo mayor en la segunda parte ya que es el uso más habitual que existe en la actualidad en los gestores de proyecto (usando actualmente servidores ftp sin almacenamiento de versiones). Resultaría positivo, en conclusión, valorar el estado de las tareas y productos de trabajo de un proyecto pudiéndose apoyar en el estado de un repositorio, y por tanto realizar un módulo sin el complementario restaría utilidad a la aplicación.

2.3.2 CVS vs SVN: Elección de Subversion

Definido el requisito de incluir un sistema de control de versiones en un gestor de proyectos (en dotproject, como se justificará más adelante), el siguiente paso fue decidir qué aplicación incluir de las que cumplían esta función. Teniendo en cuenta usar un sistema no propietario, las opciones se reducían por su popularidad y funcionamiento a CVS y Subversion.

Se realizó un estudio de ambas herramientas, percibiendo un mayor uso de CVS, pero encontrando que el mercado estaba apostando por Subversion en la actualidad ya que CVS era un proyecto que ya no estaba en desarrollo, presentaba defectos estructurales, y la gran comunidad de usuarios se basaba en ser el primer sistema de control de versiones que se popularizó con la aparición de internet. Además, uno de los requisitos

deseados era permitir implementar permisos en el repositorio basados en usuarios, tarea que en CVS es muy compleja [Fish 07]. Se aporta la siguiente comparativa realizada en la primera fase de elaboración del proyecto:



Diferencias entre CVS y SVN.

Por todo ello, se decidió el uso de Subversion como sistema de control de versiones a utilizar en el proyecto. Una vez tomada la decisión, se elaboró un estudio más detallado de las funcionalidades de Subversion, elaborando una guía rápida sobre sus opciones y comandos de uso, que se adjunta en este documento como Anexo D.

2.3.3 Gestor SVN: Necesidad de herramienta intermedia

Para integrar Subversion en una aplicación existían dos alternativas:

- Realizar llamadas a los servicios de Subversion desde el gestor de proyectos, accediendo al núcleo de Subversion para realizar las operaciones habituales.

- Utilizar una herramienta intermedia que centralizara el uso de Subversion y sirviera de intermediario entre los servicios de subversion y el gestor de proyectos.

Elegimos la segunda opción debido a que utilizar una herramienta auxiliar presentaba numerosas ventajas, principalmente:

- Favorecía una estructura modular en el proyecto.
- Se simplificaba la gestión de Subversion, sin necesidad de adentrarse en el código fuente de cada operación de Subversion.
- Se garantizaba un uso correcto de las operaciones de Subversion, evitando problemas en un módulo que requiere de una consistencia y robustez incluso superior a la habitual por su función de almacenar todos los ficheros del sistema.

Se realizó un análisis de las herramientas que cumplían esa función, centrándose en las alternativas que tenían mayor soporte y que ya estaban siendo utilizadas en otras aplicaciones, cuyos resultados pueden consultarse en el Anexo D.

2.3.4 Elección de SVNKit: Ventajas y desventajas

Estudiadas todas las alternativas, y basado en los motivos que nos llevaron a buscar una herramienta intermedia para gestionar Subversion, se eligió SVNKit basado en su robustez, sus numerosos ejemplos y su facilidad para ser incluido en nuevas aplicaciones como se demostraba en el gran número de proyectos que lo utilizaban.



SVNKit logo.

Utilizar SVNKit, no obstante, suponía permitir la coexistencia de dos servidores en la aplicación: Uno que procesara código PHP, y otro que soportara código JAVA. Por ello, y debido a que el proyecto pretende integrar toda la funcionalidad sobre un mismo entorno, esta decisión supone un sobreesfuerzo de integración, y requerirá de especial atención para el paso de información entre servidores, empleando principalmente cookies, y de distintos puertos para identificar claramente sobre cual se realizarán las operaciones en cada momento.

3. Análisis

A la hora de desarrollar el producto, efectuamos distintos análisis para determinar qué debía realizar, cómo debía hacerlo y bajo qué condiciones. Para ello, elaboramos diferentes documentos que se vieron modificados a medida que el proyecto evolucionaba para adaptarse a las necesidades reales que valorábamos en cada momento del desarrollo. La versión final de estos documentos se puede encontrar a continuación, dividida en tres secciones: requisitos funcionales, no funcionales y casos de uso.

3.1. Requisitos funcionales

A continuación se muestran los diferentes requisitos funcionales que hemos definido para nuestra aplicación. Los requisitos funcionales marcan los objetivos que nuestro sistema cumple. Son el principal referente a la hora de buscar cuáles son las características principales de nuestra aplicación. Hay que tener en cuenta que estos requisitos se apoyan sobre todos los que ofrece dotProject con su gestión de proyectos. DotProject permite la creación, modificación y seguimiento de proyectos junto a diferentes funciones de apoyo a todos estos procesos. Valoraremos únicamente qué requisitos funcionales adicionales ofrece nuestro sistema sobre todos los que ya ofrece dotProject.

- **Integración con dotproject**

Para facilitar tanto la reutilización de código como la interacción con el usuario, se opta por la modificación de dotproject en su propio código fuente para añadir todas las nuevas características. Pese a realizar un intrusismo sobre su código, estas modificaciones apenas afectan a una docena de ficheros, ya que el cuerpo principal de la aplicación está contenido dentro del módulo FVE que se instala sobre dotProject. Dotproject tiene como base de funcionamiento bases de datos MySQL, código en PHP e interfaz visual HTML (con presentación vía CSS). De esta forma, nuestra aplicación ofrecerá todas sus nuevas funciones integradas con dotProject y los usuarios interactuarán con ellas a través de su navegador Web favorito.

A pesar de que nuestro sistema añadirá nueva información que dotproject no gestiona por defecto, se respeta la estructura base utilizada por dotproject, de manera que los datos comunes entre nuestro módulo y dotproject se mantendrán sin modificación. Así, desde dotproject podremos ver elementos de nuestro módulo aunque no contengan toda la nueva información añadida. Se pretende así maximizar la compatibilidad entre ambos y, en caso de desearlo, poder realizar una instalación de nuestra aplicación sobre una versión existente de dotProject manteniendo toda la información ya contenida en él.

- **Importación de definición proyectos en XML**

Nuestro sistema permitirá a los usuarios importar un proyecto previamente definido en la aplicación EPF composer. Lo único que tenemos que hacer es una vez definido el proceso en EPF exportarlo como un archivo XML. Una vez tengamos el fichero XML en nuestro sistema de ficheros, podremos importar toda esa información en dotProject mediante un asistente que nos ayudará a ir configurando todos los parámetros que se han obtenido desde el fichero XML. Este asistente (wizard) permitirá el correcto ajuste de los datos importados del proyecto a definir en dotproject. Algunos de los parámetros que se podrán ajustar son:

1. *Información del nuevo proyecto:* descripción y nombre del proyecto así como las fechas de inicio y de fin.
2. *Asignación de un repositorio:* se permite la creación de un repositorio Subversion específico para cada proyecto para facilitar las tareas de desarrollo.
3. *Asignación de personal al proyecto:* qué usuarios participarán en el proyecto.
4. *Asignación de roles al personal del proyecto:* determinar qué roles podrán desempeñar los usuarios insertados en el proyecto.
5. *Ajuste de productos de trabajo:* qué usuarios participarán en cada producto de trabajo y qué rol desempeñarán. Se permite subir un archivo plantilla desde el disco duro del usuario a cada producto de trabajo.
6. *Ajuste de las tareas:* para cada tarea importada, se puede definir qué usuarios trabajarán en ella y sus fechas de inicio y fin.

- **Adaptación de proyectos a las funciones de FVE**

Además de realizar una importación a partir de un proyecto previamente definido en EPF, se permitirá a los usuarios la creación de proyectos de manera independiente (siguiendo el proceso estándar ofrecido por dotProject) para luego adaptar estos proyectos e introducir sobre ellos toda la nueva información que ofrece FVE. Todo esto se realiza conservando la estructura que tenía el proyecto previamente a su conversión. Igualmente, podemos crear un repositorio de Subversion sobre estos proyectos para empezar a trabajar con éste repositorio.

- **Gestión de productos de trabajo**

Una de las principales características de nuestro proyecto son los productos de trabajo. Un producto de trabajo podría definirse como el resultado material de una tarea, ya sea documentación, código fuente o únicamente

ideas conceptuales. Estos productos de trabajo podrán ser salidas o entradas de nuestras tareas y podremos asociar un archivo físico a cada uno de ellos. En nuestro proyecto se permite la gestión de todos estos aspectos de los productos de trabajo.

- **Modificación de las propiedades de un proyecto**

Los proyectos que estén en el sistema, pueden recibir modificaciones en cualquiera de sus componentes: productos de trabajo, tareas, roles, usuarios... etc. No se limita la modificación de estos atributos únicamente en los asistentes. Así, en cualquier momento, los usuarios podrán realizar las modificaciones que consideren oportunas en cualquier aspecto del proyecto, manteniendo siempre la consistencia de los citados proyectos.

- **Seguimiento de proyectos**

Tanto desde el punto de vista del jefe de proyecto como del resto de usuarios, será necesario poder realizar un seguimiento de los proyectos activos. Este seguimiento permitirá conocer el estado general del proyecto que concierne a cada usuario en función de su rol. Por ejemplo, el jefe de proyecto puede poder conocer la actividad actual del proyecto, la asignación del personal a las distintas tareas o el estado de terminación de las mismas. Para cada elemento existente en el proyecto, se puede visualizar una página en la que aparece toda su información detallada para poder valorar en cada momento cómo se encuentra la actividad concreta.

Igualmente, un usuario que esté asignado a una tarea debe conocer qué debe realizar (productos de trabajo), quiénes son los compañeros que trabajarán con él o cuáles son los plazos de terminación de la tarea.

- **Utilización de un sistema de control de versiones (SVN)**

Desde la aplicación, los usuarios podrán hacer uso de un sistema de control de versiones, más en concreto SVN (Subversion). Gracias a este sistema, el proyecto podrá tener asociado un repositorio en el cuál se ubique toda la información que los usuarios deseen asociar al proyecto (código fuente, documentación, etc.) evitando así el problema de tener información redundante con diferentes versiones en distintas máquinas.

El sistema SVN estará respaldado y protegido por un sistema basado en los permisos y roles de los usuarios del sistema. De esta forma, a la hora de trabajar con SVN, se protegerá que los usuarios sólo puedan acceder a los repositorios de los proyectos en los que estén trabajando, sin interferir así en los demás repositorios.

La interacción con el repositorio se podrá realizar a través de la propia interfaz de la aplicación, donde el usuario podrá realizar las siguientes actividades:

- Explorar el contenido con la posibilidad de descargarse los archivos uno por uno.
- Subir un archivo para que sea insertado o actualizado (commit) en el repositorio.
- Crear nuevas carpetas en el repositorio.
- Borrar cualquier elemento del repositorio sea archivo o carpeta. Si se desea borrar una carpeta no vacía, esta será suprimida junto con su contenido.
- Descargarse una copia de trabajo del repositorio en bajo un fichero zip.
- Subir varios archivos y carpetas a la vez para su actualización o inserción mediante la subida al servidor del contenido comprimido en formato zip.
- Obtener la información que proporciona subversion sobre el repositorio (versión, autor, última fecha de actualización, etc.).

- **Realización de informes adaptados de los proyectos**

Para los diferentes proyectos que tengamos instanciados en la aplicación, podremos realizar, a parte de los informes que ya ofrece dotProject, un nuevo tipo de informe en el que se incluye toda la nueva funcionalidad que ofrece nuestro sistema: productos de trabajo, usuarios, roles, etc. De esta forma se puede valorar de manera general el estado de nuestro proyecto con todas las nuevas características que ofrece FVE.

3.2. Requisitos no funcionales

Los requisitos no funcionales son los que marcan el marco de ejecución de nuestro sistema para que éste se ejecute de forma adecuada en términos de velocidad, presentación, etc. Imponen restricciones en el diseño o la implementación, ya que su cumplimiento es muy importante de cara a la calidad del producto frente al usuario. Nuestra aplicación se ejecuta en un navegador web ya que su interfaz es puramente HTML, así que esto condiciona los requisitos no funcionales.

Hay que tener en cuenta que para hacer funcionar la aplicación se precisa de varios servicios instalados en el ordenador huésped. Estos servicios no tienen una carga de proceso elevada, por lo que fácilmente serán ejecutables en prácticamente la mayoría de máquinas actuales. Sin embargo, los usuarios accederán a este servidor vía web a través de un navegador web, por lo que los requisitos necesarios para funcionar con fluidez deberán ser aún menores ya que no existe ningún proceso que deba realizarse en sus equipos salvo la interpretación del código HTML por parte del navegador web. Para ver más información del diagrama de despliegue, ir a la sección 4.3.1.

Es preciso antes de definir los requisitos no funcionales hacer una mención a todas las aplicaciones que se requieren para poder hacer funcionar FVE. El equipo que albergue FVE debe tener activos una serie de servicios para poder funcionar correctamente. Estos servicios son:

- Servidor web Apache con funciones de PHP activas.
- Servidor de base de datos MySQL (éste puede ser incluso externo al servidor que albergue la aplicación).
- Servidor de aplicaciones Tomcat.
- Servidor de Subversion.

Así pues y conociendo ya qué se necesita tener disponible en servidor, los requisitos no funcionales que hemos definido en nuestra aplicación son los siguientes:

Nombre	RNF-1: Ejecución en sistemas operativos Linux y Windows
Requisitos asociados	-
Descripción	La aplicación debe poderse ejecutar tanto en sistemas operativos Linux como Windows.
Importancia	Alta: debido a la temática de la aplicación, ésta podría ser desplegada en muy distintos ambientes: empresariales, educativos, personales... Por esta razón, dar soporte a todos los sistemas operativos fue una prioridad a la hora de desarrollar el programa.
Estado	Probado bajo Windows XP SP2 y en Linux con distribución Fedora.
Comentarios	Debido a que los servicios necesarios por la aplicación (Apache, MySQL y Tomcat) tienen versiones tanto para Windows como para Linux es posible desplegar la aplicación en ambos sistemas operativos sin ningún tipo de problema.

Nombre	RNF-2: Velocidad de respuesta del sistema adecuada
Requisitos asociados	RNF-4
Descripción	<p>Debido a que FVE se trata de una aplicación basada en interfaz de usuario, la velocidad con la que las diferentes páginas se cargan debe ser adecuada para que el usuario no se sienta incómodo mientras esté trabajando. Este interfaz de usuario se presenta mediante una página web HTML que será la que visualicen los usuarios que trabajen con FVE. Para lograr conseguir esta fluidez de recarga de las distintas páginas, hay que tener en cuenta varios factores:</p> <ul style="list-style-type: none"> - Prestaciones del ordenador que esté ejecutando la aplicación: a mayor velocidad de procesador, mayor tamaño de memoria RAM, se logrará mayor velocidad de ejecución de la aplicación. De igual forma, este equipo deberá contar con espacio de almacenamiento de sobra para almacenar toda la información de los proyectos que gestione (ya que estos subirán información al repositorio Subversion). - Prestaciones del ordenador del usuario que se conecta con FVE: este equipo no necesita tener unas características especiales, únicamente un sistema operativo con un navegador web que funcione con soltura. - Ancho de banda de la conexión del ordenador servidor: el ordenador servidor de FVE deberá responder a las peticiones que realicen todos los usuarios que estén conectados desde sus ordenadores. Estos usuarios pueden ser desde uno a múltiples, por lo que la conexión debe tolerar con holgura el tráfico de datos existente entre servidor y usuario. - Ancho de banda de la conexión del usuario: este usuario se conectará con el servidor que alberga la aplicación y pese a no necesitar tanto ancho de banda como el otro lado de la conexión, si precisará de al menos cierto margen para poder trabajar con soltura y subir y bajar archivos con una tasa de datos aceptable. <p>Así pues, los anteriores factores determinaran en un grado importante el funcionamiento de la aplicación de cara al usuario, aunque siempre existen factores relevantes que pueden condicionar el rendimiento de la aplicación que es muy difícil de catalogar: procesos activos en el equipo servidor o tamaño de los archivos con</p>

	los que se trabaja.
Importancia	Alta: el rendimiento de la aplicación es muy importante, además en un gestor de proyectos en el que hay una interacción continua con los usuarios es muy importante que esta respuesta sea rápida y efectiva, porque si no podría generar incomodidad en el usuario a la hora de trabajar. Una respuesta por debajo de los 5 segundos para la carga de páginas web es adecuada y no causa incomodidad a la hora de navegar por la interfaz. Valores inferiores a los 5 segundos son mejores.
Estado	Probado en un servidor Linux de gama media para 10 accesos a diferentes páginas del interfaz de la aplicación. Se logró un tiempo medio de acceso de 1.606 segundos.
Comentarios	Pese a parecer que la aplicación tiene requisitos exigentes, en realidad funciona con bastante fluidez en equipos de gama baja ya que los servicios que utiliza (Apache, MySQL, etc.) están muy optimizados y no tienen una carga de proceso elevada.

Nombre	RNF-3: Ejecución tanto en LAN como en Internet
Requisitos asociados	-
Descripción	<p>A la hora de desplegar la aplicación, esta puede estar orientada a utilizarse en una red de área local o a través de internet.</p> <p>Ambas alternativas son iguales de cara a la configuración del servidor, sin embargo, si queremos que los usuarios accedan a través de internet, es posible que sea necesario de configurar los algunos elementos (como routers o firewalls) para que se puedan realizar accesos a través de Internet.</p>
Importancia	Media: en la mayoría de los casos no existirá problema a la hora de configurar su red para lograr que los usuarios se conecten a la aplicación.

Estado	Probado tanto en LAN (red de área local) como a través de Internet.
Comentarios	-

Nombre	RNF-4: Respuesta frente a múltiples usuarios simultáneamente sin mermar el rendimiento.
Requisitos asociados	RNF-2
Descripción	Cuando varios usuarios realicen un acceso al servidor donde está ubicada la aplicación, el rendimiento que aprecie cada usuario de manera independiente debería ser igual que cuando únicamente él está conectado.
Importancia	Media: es normal que si existe mucho tráfico de datos el servidor no responda de igual forma. Este comportamiento es comprendido por parte de los usuarios.
Estado	Probado para 4 usuarios realizando diferentes tareas sobre la aplicación simultáneamente sin apreciar deterioro del rendimiento del programa.
Comentarios	Salvo que los usuarios realicen operaciones que consuman un mayor ancho de banda (como subidas o bajadas de archivos), el rendimiento de la aplicación es bueno hasta un número aceptable de usuarios.

Nombre	RNF-5: Acceso a la aplicación desde diferentes navegadores web.
Requisitos asociados	-

Descripción	El usuario que utilice la aplicación deberá poder acceder a la aplicación utilizando su navegador web favorito.
Importancia	Alta: dando soporte a diferentes navegadores se consigue que el usuario pueda emplear el navegador con el que esté acostumbrado a trabajar para acceder a nuestra aplicación.
Estado	Probado con Firefox, Internet Explorer 7, Safari, Ice Weasel.
Comentarios	-

3.3. Casos de uso

A continuación se exponen todos los casos de uso que implementa nuestro sistema. En ellos se pretende dar una visión del funcionamiento de la aplicación proporcionando uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Es importante indicar que en los siguientes casos de uso, se asumen las siguientes condiciones para evitar tener que replicarlas en cada uno:

- El usuario ha iniciado sesión en la aplicación, ya sea como administrador o como usuario estándar.
- El módulo FVE Project Manager debe estar activo en dotProject.

Listado de casos de uso:

En el siguiente listado previo se muestra todos los casos de uso que reflejan el comportamiento de nuestra aplicación. Están organizados en distintos niveles para facilitar su organización y entendimiento.

1. Creación de un proyecto FVE

1.1. Instanciar un proyecto a partir del Wizard de creación de proyectos.

1.2. Crear un proyecto y convertirlo a formato FVE

1.3. Transformar una tarea estándar de dotProject a una tarea con elementos de EPF.

2. Ejecución de un proyecto FVE

2.1. Visualizar información del propio proyecto y sus tareas

2.1.1. Generar un informe del estado de un proyecto con sus elementos EPF.

- 2.1.2. Visualizar todas las tareas EPF en las que participa un usuario.
- 2.1.3. Visualizar la información EPF de un proyecto.
- 2.1.4. Visualizar la información EPF de una tarea.

2.2. Visualización de roles

- 2.2.1. Visualizar los roles que intervienen en un proyecto FVE
- 2.2.2. Visualizar los roles que intervienen en cualquier proyecto FVE del sistema
- 2.2.3. Ver información detallada de un rol EPF
- 2.2.4. Visualizar todos los roles que puede desempeñar un usuario en los proyectos FVE.
- 2.2.5. Visualizar todos los roles que desempeña un usuario dentro de los proyectos FVE en los que participa.

2.3. Ejecución de productos de trabajo

- 2.3.1. Modificar el progreso de un producto de trabajo.
- 2.3.2. Visualizar los productos de trabajo incluidos dentro de un proyecto FVE
- 2.3.3. Visualizar todos los productos de trabajo incluidos en cualquier proyecto FVE.
- 2.3.4. Ver información detallada de un producto de trabajo.
- 2.3.5. Visualizar todos los productos de trabajo en los que participa un usuario.

2.4. Instanciación de Plantillas

- 2.4.1. Visualizar todas las plantillas asociadas a los productos de trabajo de un proyecto EPF.
- 2.4.2. Visualizar todas las plantillas de cualquier proyecto EPF.
- 2.4.3. Ver información detallada de una plantilla.
- 2.4.4. Descarga del archivo asociado a la plantilla.

2.5. Ejecución del repositorio de Subversion

- 2.5.1. Acceder al panel de gestión de subversion asociado a un proyecto.
 - 2.5.1.1. Visualización de la información del repositorio de un proyecto
 - 2.5.1.2. Obtención de la copia de trabajo del repositorio asociado a un proyecto
 - 2.5.1.3. Despliegue y visualización del contenido del repositorio a través de su árbol de exploración
 - 2.5.1.4. Acceder al gestor de archivos de subversion "File Manager"
 - 2.5.1.4.1. Navegación a través del repositorio mediante el uso del gestor de archivos (File Manager)

3. Modificación de un proyecto FVE

- 3.1. Modificación de parámetros de un proyecto y sus tareas
 - 3.1.1. Editar la información EPF de un proyecto.

- 3.1.2. Editar los usuarios que pueden trabajar en un proyecto FVE.
 - 3.1.3. Editar la información EPF de una tarea.
 - 3.1.4. Borrar un proyecto FVE.
 - 3.1.5. Borrar una tarea EPF.
 - 3.2. Modificación de roles de un proyecto
 - 3.2.1. Borrar un rol EPF.
 - 3.2.2. Editar un rol EPF (nombre, descripción y usuarios que lo cumplen).
 - 3.2.3. Crear un rol EPF.
 - 3.3. Modificación de los productos de trabajo de un proyecto
 - 3.3.1. Borrar un producto de trabajo.
 - 3.3.2. Editar información general de un producto de trabajo.
 - 3.3.3. Editar información de roles y tareas de un producto de trabajo.
 - 3.3.4. Editar información de usuario responsable y usuarios modificadores de un producto de trabajo.
 - 3.3.5. Crear un producto de trabajo.
 - 3.4. Modificación de una plantilla de un proyecto
 - 3.4.1. Borrar una plantilla.
 - 3.4.2. Editar la información de una plantilla (nombre, producto de trabajo y fichero asociado).
 - 3.4.3. Crear una plantilla.
 - 3.5. Modificación del repositorio de Subversion
 - 3.5.1. Importar los archivos de un proyecto al repositorio asociado.
 - 3.5.2. Realización de una actualización del repositorio asociado a un proyecto (Commit) Paso 1
 - 3.5.3. Realización de una actualización del repositorio asociado a un proyecto (Commit) Paso 2
 - 3.5.4. Borrar archivos o carpetas mediante el gestor de archivos de Subversion "File Manager"
 - 3.5.5. Añadir archivos al repositorio desde el gestor de archivos de Subversion "File Manager"
 - 3.5.6. Crear nuevas carpetas en el repositorio usando el gestor de archivos de Subversion "File Manager"
 - 4. Modificación de los parámetros del módulo de Subversion
 - 4.1. Acceder al panel de configuración de los componentes de Subversion
 - 4.2. Modificar el estado de subversion
 - 4.2.1. Modificar la cadena de enrutamiento de subversion
 - 4.2.2. Modificar el puerto de enrutamiento de subversion
 - 4.3. Creación del repositorio local

A continuación ya procedemos a mostrar los casos de uso en detalle.

3.3.1. Creación de un proyecto FVE

Caso de uso 1	Creación de un proyecto FVE
Nivel de abstracción	1
Objetivos	Instanciar un nuevo proyecto en el sistema para poder poner a los usuarios a trabajar en él.
Actores	Cliente vía web, sistema.
Entradas	Información del proyecto que se desea instanciar: nombre, descripción, archivo XML creado con EPF Composer con la definición del proyecto.
Condiciones previas	Conocer qué proyecto se desea instanciar con toda su información: usuarios, tareas, roles, plazos de inicio y finalización, etc.
Salidas	Un nuevo proyecto instanciado en el sistema en el que los usuarios podrán ponerse a trabajar.
Post condición si éxito	El sistema configurado de forma coherente con un nuevo proyecto con sus tareas, usuarios, roles, etc. Este proyecto estará instanciado para poder comenzar a trabajar en él inmediatamente.
Post condición si fallo	El proyecto no se crea en el sistema y el usuario es informado de que no se ha podido instanciar su proyecto.
Activador	Creación de un proyecto a partir del wizard de creación de proyectos. Crear un proyecto de forma tradicional y adaptarlo al formato FVE.
Frecuencia de uso	Media, la instanciación de proyectos es precisa realizarla una única vez por proyecto. Sin embargo, si se utilizan muchos proyectos en el sistema, siempre es preciso haber pasado por esta etapa.

Caso de uso 1.1	Instanciar un proyecto a partir del Wizard de creación de proyectos
Nivel de abstracción	2 Caso de uso padre: 1 - Creación de un proyecto FVE.
Objetivos	Crear un nuevo proyecto EPF en el que podemos importar la información desde un archivo XML creado con EPF Composer y que además se permite ajustar la información importada para adaptarla a nuestras necesidades. Todo esto se realiza a través de un asistente que en varios pasos va guiando al usuario a realizar las operaciones citadas anteriormente. Se puede indicar al sistema que asocie un repositorio SVN (Subversion) al nuevo proyecto.
Actores	Cliente vía web, sistema.
Entradas	Información de proyecto: descripción, nombre, fechas inicio y fin. Archivo XML con la estructura base del proyecto, exportado con EPF Composer. Información sobre el repositorio del proyecto.

	<p>Usuarios que participarán en el proyecto.</p> <p>Asignación de usuarios a los roles que se han importado desde el archivo XML del proyecto.</p> <p>Parámetros de los productos de trabajo importados. Se permite subir archivos asociados como plantillas a los productos de trabajo.</p> <p>Parámetros de las tareas.</p>
Condiciones previas	Únicamente haber iniciado sesión en el sistema.
Salidas	Mensaje de confirmación de que el nuevo proyecto ha sido creado con éxito.
Post condición si éxito	Se crea un nuevo proyecto en el sistema y sus parámetros quedan configurados en base a lo que el usuario introdujo en el asistente.
Post condición si fallo	Se muestra por pantalla un mensaje de error y el proyecto ni ninguno de sus elementos (tareas, usuario, roles) se crea.
Activador	<p>Botón "New EPF Project Wizard" en la pantalla "Projects".</p> <p>Botón "Launch Project Creation Wizard" en la parte "Projects" del módulo FVE Project Manager.</p>
Secuencia	<p>0. El usuario lanza el asistente de creación de proyectos. En cualquier momento, puede pulsar los siguientes botones:</p> <p>Pulsar botón "Pause Wizard": Ir a 9.</p> <p>Pulsar botón "Cancel Wizard": Una ventana desplegable pregunta al usuario si desea cancelar el proyecto. Si la respuesta es "Aceptar", ir a 10. Si no, se permanece en el mismo punto del asistente.</p> <p>Al arrancar el wizard, si existían una sesión anterior de wizard en estado de pausa, se retomará por el punto en el que se dejó y se cargarán todos los datos que el usuario había introducido.</p> <p>Si no existía una sesión de wizard en pausa, se lanza el wizard en el primer paso, ir a 1.</p> <p>1. El usuario debe introducir nombre de proyecto, descripción y fechas de inicio y finalización. También el propietario del proyecto. Si pulsa el botón del calendario para seleccionar las fechas, ir a 11. Si pulsa botón "Next step" ir a 2.</p> <p>2. Debe elegirse un archivo XML si se desea importar un esquema de proyecto creado con EPF Composer. Para ello, marcar el checkbox y después pulsar el botón "Examinar" para elegir el archivo XML del que se desea importar. También se puede elegir si se quiere utilizar un repositorio o no. Si se pulsa "Next step" ir a 3, si se pulsa "Previous step" ir a 1.</p> <p>2A: Es la pantalla de importación en la que ya se había importado previamente un archivo XML. Si se desea conservar los cambios como hasta ahora, desmarcar la casilla "Do you wish to import from an EPF XML file?". Si se desea cambiar el archivo XML que se importó, pulsar el botón "Examinar" y elegir un nuevo fichero XML válido. Pulsar "Next step" (ir a 3) o "Previous Step" (ir a 1). Si se produce un error al importar ir a E-1 (Excepciones).</p> <p>3. Configuración de usuarios. Se debe elegir de la lista desplegable que contiene todos los usuarios existentes en el sistema los que</p>

	<p>queremos que participen en el proyecto. Pulsar el botón "Add user" e ir añadiendo usuarios. Pulsar "<i>Next step</i>" (ir a 4) o "<i>Previous Step</i>" (ir a 2).</p> <p>4. Configurar roles. Por cada rol importado desde el archivo XML, podremos insertar en él desde la lista de usuarios que configuramos en el paso 2 para que participen en el sistema. Pulsar "Add user to role" en el rol correspondiente para añadir este usuario al rol. Cuando hayamos terminado, pulsar "<i>Next step</i>" (ir a 5) o "<i>Previous Step</i>" (ir a 3).</p> <p>5. Configurar Productos de Trabajo. Por cada producto de trabajo importado desde el archivo XML, podemos configurar su usuario responsable (únicamente uno) y sus usuarios modificadores (cero o más por cada rol). Los usuarios que configuremos deben cumplir el rol especificado para el puesto que ocupen. También podemos asociar un archivo de plantilla al producto de trabajo. Para ello, marcar el check "Do you want to associate a template file for this workproduct?" y después el botón "Examinar" para elegir un archivo cualquier de nuestro sistema de archivos. También podemos configurar nombre y descripción del template. Cuando hayamos terminado, pulsar "<i>Next step</i>" (ir a 6) o "<i>Previous Step</i>" (ir a 4).</p> <p>5A: Configuración de Productos de trabajo cuando ya se ha insertado previamente una plantilla a un producto de trabajo. Aparecerá el checkbox de la plantilla marcado y se visualizará una tabla con el nombre de la plantilla previamente subida. Si se desea borrar, desmarcar el checkbox. Si se desea subir otro archivo, pulsar el botón "Examinar" y elegir el nuevo archivo. También podemos modificar su nombre o su descripción modificando las casillas "Name" y "Description". Cuando hayamos terminado, pulsar "<i>Next step</i>" (ir a 6) o "<i>Previous Step</i>" (ir a 4).</p> <p>6: Configuración de tareas. Por cada tarea importada desde el archivo XML podemos configurar su usuario primario (únicamente uno) y sus usuarios adicionales (cero o más por cada rol). Los usuarios que configuremos deben cumplir el rol especificado para el puesto que ocupen. También podemos configurar sus fechas y horas de inicio y fin. Para elegir una fecha, pulsar el botón del calendario (ir a 11). Si deseamos saber el tiempo real de trabajo de la tarea, pulsar el botón "Calculate Duration" y éste se mostrará en el cuadro "Duration". Cuando hayamos configurado las tareas acorde a nuestras necesidades, pulsar "<i>Next step</i>" (ir a 5) o "<i>Previous Step</i>" (ir a 7).</p> <p>7: Confirmación de creación de proyecto: este es el último paso antes de finalizar el wizard por si se desean realizar cambios sobre él. Si queremos repasar algo, pulsar "Previous step" (ir a 6). Si estamos satisfechos con la configuración realizada, pulsar Finish (ir a 8).</p> <p>8: Todos los cambios quedarán almacenados y se retornará a la página <i>Projects</i> de dotProject donde podremos ver creado nuestro proyecto.</p> <p>9: El proyecto queda en estado de stand-by hasta la próxima vez que el usuario lo lance. Durante este tiempo, los cambios realizados quedan almacenados temporalmente.</p> <p>10: El proyecto se cancela, borrando toda la información que se había creado en relación a él. Se sale del asistente volviendo a la</p>
--	---

	<p>pantalla.</p> <p>11: Surge una ventana que permite al usuario elegir una fecha en el calendario que muestra. Al pulsar una fecha, ésta se modificará en el parámetro que tenía asociado y se volverá al paso del wizard en el que se llamó.</p>
Excepciones	E-1: El sistema intentará importar el archivo XML y se producirá un error. Esta circunstancia se indicará por pantalla.
Frecuencia de uso	Media/Alta: cada vez que deseemos instanciar un proyecto a partir de un archivo XML deberemos hacer uso de este asistente.

Caso de uso 1.2	Crear un proyecto y convertirlo a formato FVE
Nivel de abstracción	2 Caso de uso padre: 1 - Creación de un proyecto FVE.
Objetivos	<p>Crear un proyecto utilizando el método estándar de creación de proyectos facilitado por dotProject para luego introducir la información adicional que facilita FVE Project Manager.</p> <p>Este método también permite adaptar un proyecto existente al formato utilizado por FVE añadiendo toda la nueva funcionalidad que éste presta.</p>
Actores	Cliente vía web, sistema.
Entradas	Datos del nuevo proyecto que se quiere instanciar.
Condiciones previas	Conocer qué proyecto se desea instanciar con toda su información: usuarios, tareas, roles, plazos de inicio y finalización, etc.
Salidas	Información de EPF asociada al nuevo proyecto cuando lo visualizamos en la pantalla "Projects" de dotProject.
Post condición si éxito	Un nuevo proyecto creado en el con información EPF asociada a él en el que los usuarios pueden empezar a trabajar.
Post condición si fallo	El nuevo proyecto no se instancia y se avisa al usuario de que se ha producido un error.
Activador	<p>Botón "New Project" en el módulo "Projects" de dotProject para instanciar un nuevo proyecto estándar.</p> <p>Botón "EPF Project Info" sobre la página de información del proyecto para añadir información EPF extra al proyecto convirtiéndolo así en un proyecto FVE.</p>
Secuencia	<p>1. En primera instancia hay que crear un proyecto de igual forma que se hacía con dotProject sin la extensión FVE. Para ello vamos a la página de dotProject y pulsamos el botón "new Project". Ir a 2.</p> <p>2. En la pantalla de creación de proyecto, introducimos toda la información relativa al proyecto y cuando hayamos finalizado, pulsar "submit". Ir a 3.</p> <p>3. Nuestro nuevo proyecto se habrá creado y aparecerá listado en la lista de proyectos. Hacemos click sobre su nombre en la lista de</p>

	<p>proyectos y pasamos al siguiente paso (ir a 4).</p> <p>4. Al entrar en la pantalla de información del proyecto, podemos apreciar que no tiene información EPF asociada (<i>Doesn't have EPF info</i>) en el cuadro de color amarillo. Para poder añadir a nuestro proyecto información EPF, pulsamos el botón "EPF Project Info". Saltamos al paso 5.</p> <p>5. Para editar la información EPF, podemos modificar varios cuadros de texto donde podemos añadir la información EPF que deseemos, tal como el enlace con la información del proyecto (<i>Guideline Link</i>) o el tipo de plugin implementado (<i>Plugin Name</i>). Cuando estemos contentos con los cambios, pulsamos el botón "Edit Project" (ir a 6). Si deseamos descartar los cambios, pulsar el botón "Back" (ir a 6.A).</p> <p>6.A. Volveremos a la pantalla de nuestro proyecto donde veremos que el proyecto sigue siendo de tipo estándar y no se han aplicados los cambios que estábamos realizando en la pantalla de edición de información EPF.</p> <p>6. Una vez editada la información de EPF del proyecto, podemos editar los usuarios que forman parte de este proyecto. Para más información, mirar el caso de uso 3.1.2. Cuando estamos satisfechos con los cambios, pulsar el botón "Finish EPF Editing" (ir a 7).</p> <p>7. Tras haber realizado la modificación del proyecto, se mostrará una pantalla en la que se informará de que toda la operación de edición se ha realizado con éxito. Pulsamos el texto "Go back to Project info page" y volveremos a la pantalla de información de nuestro proyecto (ir a 8).</p> <p>8. De nuevo nos encontraremos en la página de información de nuestro proyecto, pero ahora en el recuadro de información EPF (recuadro amarillo) podemos ver patentes los cambios que hemos realizado sobre el proyecto. Ahora nuestro proyecto es del tipo FVE, con toda la nueva funcionalidad que ofrece nuestro módulo.</p>
Excepciones	-
Frecuencia de uso	Baja/Media: la forma más común de instanciar un proyecto es utilizando el wizard de creación de proyectos (Caso de uso XX), pero si se desea modificar un proyecto existente o crear un nuevo mediante el método tradicional para luego agregarle toda la funcionalidad EPF, este caso de uso puede ser muy útil.

Caso de uso 1.3	Transformar una tarea estándar de dotProject a una tarea con elementos de EPF
Nivel de abstracción	2 Caso de uso padre: 1 - Creación de un proyecto FVE.
Objetivos	Las tareas por defecto que se crean en nuestro proyecto no tienen asociada información de EPF. Para poder dar soporte a esta funcionalidad de FVE, es necesario insertar información EPF en la tarea.
Actores	Cliente vía web, sistema.

Entradas	Información EPF que queremos asociar a la tarea.
Condiciones previas	La tarea debe de estar creada en el sistema de forma correcta.
Salidas	La tarea quedará modificada en la base de datos y podremos observar su nueva información EPF en la pantalla de información de la tarea.
Post condición si éxito	Se creará la información EPF asociada a la tarea y el usuario podrá hacer uso de ella de cara a una mejor gestión de su proyecto. El resto de información de la tarea se mantendrá (fechas de inicio y fin, nombre, descripción, etc.).
Post condición si fallo	La tarea permanecerá con su información
Activador	Botón "EPF Task Info" en la pantalla de información de la tarea.
Secuencia	<p>0. En primer lugar debemos tener creada una tarea en el sistema que no tenga asociada información de EPF para poder asociársela. Para ello, nos dirigimos a su pantalla de información (haciendo click sobre ella en la información del proyecto en el que esté ubicada o en el mismo módulo Tasks de dotProject). Una vez estemos visualizando la tarea, se puede apreciar que no tiene información EPF porque aparece el texto "Add EPF task Info". Pulsar el botón "EPF Task Info" para añadirla (Ir a 1).</p> <p>1. Ahora debemos configurar la nueva información de la tarea. En este primer paso debemos configurar el tipo de tarea. Introducirlo y pulsar el botón "Next Step" (ir a 2). Si deseamos cancelar el proceso, pulsar el botón "Back" (ir a 2.A).</p> <p>2.A. Se ha optado por no introducir información EPF en la tarea y se retorna a la página de información de la tarea.</p> <p>2. Una vez introducido el tipo de tarea, debemos configurar los roles de los usuarios que serán primarios en esta tarea (un único rol) y adicionales (cero o más roles). También podemos configurar los workproducts relacionados con esta tarea y de qué tipo es la relación. Si deseamos volver al paso anterior y descartar los cambios, pulsar el botón "Previous step" (ir a 1). Si lo que queremos es confirmar los cambios, pulsar "Next Step" (ir a 3).</p> <p>3. Ahora es necesario configurar los usuarios que participaran en la tarea. Se asignarán en función de los roles que asignamos en el paso anterior. Cuando hayamos terminado, pulsamos el botón "Finish" (ir a 4) para terminar la edición de las nuevas características EPF de la tarea. Si deseamos volver al paso anterior descartando los cambios actuales, pulsar el botón "Previous Step" (ir a 2).</p> <p>4. Ya hemos terminado la edición de las nuevas propiedades EPF de la tarea. En esta pantalla se nos mostrará un mensaje de confirmación. Para regresar a la pantalla de información de la tarea, pulsar el botón "Go back to task info" (ir a 5).</p> <p>5. De nuevo estamos en la tarea de información de la tarea, pero esta vez podemos visualizar en el recuadro amarillo de "EPF Info" la nueva información EPF que hemos creado en la tarea.</p>

Frecuencia de uso	Alta. Cada vez que creemos una tarea en el sistema que no haya sido importada desde el archivo XML, si queremos hacer uso de sus capacidades EPF es necesario realizar este proceso.
-------------------	--

3.3.2 Ejecución de un proyecto FVE

Caso de uso 2	Ejecución de un proyecto FVE
Nivel de abstracción	1
Objetivos	Ejecutar un proyecto es sinónimo de poner a sus usuarios a trabajar en sus tareas e ir realizando progresos hasta lograr la terminación de los objetivos marcados en el proyecto. Para llevar a cabo esta ejecución, necesitaremos de varias herramientas (aparte de las propias que nos facilita dotProject) para poder finalizar exitosamente el proceso iniciado. Esas herramientas quedan recogidas dentro de este caso de uso: ejecución de un proceso.
Actores	Cliente vía web, sistema.
Entradas	El propio proyecto que queremos ejecutar
Condiciones previas	Tener un proyecto instanciado y correctamente configurado.
Salidas	Modificación/mejora del proyecto que se está ejecutando.
Post condición si éxito	El proyecto avanza hacia su terminación.
Post condición si fallo	El proyecto no se modifica.
Activador	-
Frecuencia de uso	Muy alta: la ejecución de un proyecto es la clave de un gestor de proyectos. Las utilidades que nuestro gestor brinda de cara a la ejecución (visor de elementos, modificar estado de tareas y productos de trabajo, etc.) son clave para poder finalizar los objetivos marcados dentro del proyecto.

Caso de uso 2.1	Visualizar información del propio proyecto y sus tareas
Nivel de abstracción	2 Caso de uso padre: 2 – Ejecución de un proyecto FVE
Objetivos	Es imprescindible en todo momento conocer qué características tiene el proyecto y las tareas que éste contiene. Necesitamos conocer la información propia del proyecto y las tareas así como la información EPF que pueden tener ambos asociada.

Actores	Cliente vía web, sistema.
Entradas	El proyecto del que queremos recopilar información.
Condiciones previas	Instanciación del proyecto con todos sus parámetros correctamente definidos.
Salidas	Información pedida al sistema sobre el propio proyecto o sus tareas.
Post condición si éxito	Se suministra al usuario una información que éste puede utilizar para avanzar el proyecto hacia su terminación.
Post condición si fallo	No se consigue ninguna información del proyecto.
Activador	Diversos enlaces repartidos por las páginas de los proyectos y tareas.
Frecuencia de uso	Muy alta. Consultar la información del proyecto

Caso de uso 2.1.1	Generar un informe del estado de un proyecto con sus elementos EPF
Nivel de abstracción	3 Caso de uso padre: 2.1 – Visualizar información del propio proyecto y sus tareas.
Objetivos	Obtener por pantalla un informe que incluya información sobre los nuevos elementos EPF que intervienen en un proyecto.
Actores	Cliente vía web, sistema.
Entradas	El proyecto del que deseamos realizar el informe.
Condiciones previas	Tener un proyectos instanciado y configurado que tenga algún elemento EPF.
Salidas	Un informe visual en el que se incluye información de terminación de los productos de trabajo de un proyecto FVE, usuarios que trabajan en el proyecto con su asignación de trabajo y otra información útil.
Post condición si éxito	Se muestra por pantalla (con posibilidad de impresión) el informe con los datos EPF del proyecto (junto a su información general).
Post condición si fallo	No se muestra el informe.
Activador	Enlace "Reports" en la parte superior de la pantalla de información de un proyecto. Una vez estemos en la pantalla de Reports, elegir el informe "Project Statistics: Include FVE Stats".
Secuencia	<p>1. Acceder a la pantalla de informes de un proyecto pulsando el botón "Reports" en la parte superior de su pantalla de información general. Pasar al punto 2.</p> <p>2. Elegir el informe que incluye información de las características nuevas incluidas en la aplicación: "Project Statistics: Include FVE"</p>

	<p><i>Stats</i>". Pasar al punto 3.</p> <p>3. Se mostrará por pantalla en una nueva página el informe que contiene información de los productos de trabajo, carga de trabajo de los usuarios, etc.</p>
Excepciones	3.B. Si el proyecto aún no tiene elementos EPF, simplemente estos parámetros se obviarán a la hora de presentar el informe.
Frecuencia de uso	Baja. Es una función muy útil, pero su utilización está limitada a puntos muy concretos de la ejecución del proyecto.

Caso de uso 2.1.2	Visualizar todas las tareas EPF en las que participa un usuario
Nivel de abstracción	3 Caso de uso padre: 2.1 – Visualizar información del propio proyecto y sus tareas.
Objetivos	Conocer en cuántas tareas EPF trabaja un usuario y a qué proyectos pertenecen.
Actores	Cliente vía web, sistema.
Entradas	Un usuario del sistema cualquiera.
Condiciones previas	El usuario debe estar creado en la aplicación de forma correcta.
Salidas	Una lista de tareas organizadas por el proyecto al que pertenecen. Para cada tarea se muestra qué papel tiene el usuario en ella (rol y relación con la tarea: primario o modificador).
Post condición si éxito	Se muestra por pantalla el informe con las tareas en las que trabaja el usuario.
Post condición si fallo	No se muestra por pantalla ningún informe.
Activador	Pestaña "EPF Tasks" dentro de la pantalla de información de un usuario.
Secuencia	<p>1. Acceder a la pantalla de información del usuario del que deseamos conocer en qué tareas trabaja. Para acceder a esta pantalla, o bien vamos al módulo "User Admin" de dotProject o pulsamos sobre el nombre del usuario en cualquier punto de la aplicación donde éste aparezca. Pasamos al paso 2.</p> <p>2. Una vez estemos en la pantalla de información del usuario, debemos pulsar sobre la pestaña "EPF Tasks". Ir a 3.</p> <p>3. Sobre la misma página, se cambiará el contenido del cuadro de la parte inferior de la pantalla para mostrar el informe con las tareas en las que éste usuario interviene.</p>
Excepciones	-
Frecuencia de uso	Baja/Media: puede ser útil conocer durante el desarrollo de un proyecto la asignación de tareas de un usuario, pero no es una

	acción que ejecute de forma repetida.
--	---------------------------------------

Caso de uso 2.1.3	Visualizar la información EPF de un proyecto
Nivel de abstracción	3 Caso de uso padre: 2.1 – Visualizar información del propio proyecto y sus tareas.
Objetivos	Comprobar la información EPF asociada a un proyecto.
Actores	Cliente vía web, sistema.
Entradas	Proyecto del que deseamos conocer la información EPF relacionada.
Condiciones previas	Tener el proyecto del que queremos conocer la información instanciado correctamente en el sistema.
Salidas	Recuadro en el que aparece toda la información EPF relacionada del proyecto: Ruta del repositorio SVN, enlace a la información EPF del proyecto y plugin utilizado. También podremos ver qué usuarios trabajan en el proyecto.
Post condición si éxito	Mostrar la información pedida al usuario.
Post condición si fallo	No se muestra la información.
Activador	Cualquier enlace que lleve a la pantalla de información del proyecto. Generalmente basta con hacer click sobre el nombre del proyecto en cualquier página de la aplicación donde éste aparezca. Si no, siempre en el módulo “Projects” de dotProject podremos acceder a todos los proyectos existentes en el sistema.
Secuencia	1. Hacer click sobre el nombre del proyecto en cualquier punto indicado en la sección “Activador”. Si el proyecto tiene información EPF ya creada, ir a 2A. Si no, ir a 2B. 2A. En este punto estaremos en la página de información del proyecto. Podremos apreciar por el centro de la página que hay un recuadro amarillo. Ahí estará contenida la información EPF relacionada con el proyecto. 2B. Al entrar en la página de información de proyecto, apreciaremos que sobre el centro de la página aparecerá un recuadro amarillo con el texto “Doesn't have EPF info”. Esto indica que el proyecto aún no tiene información EPF asociada. Si se desea crear, mirar el caso de uso 1.2 .
Excepciones	-
Frecuencia de uso	Media/Baja, ya que la visualización de la información EPF es algo importante pero no de uso excesivo.

Caso de uso 2.1.4	Visualizar la información EPF de una tarea
Nivel de abstracción	3 Caso de uso padre: .2.1 – Visualizar información del propio proyecto y sus tareas.
Objetivos	Comprobar la información EPF asociada a una tarea.
Actores	Cliente vía web, sistema.
Entradas	Tarea de la que deseamos conocer la información EPF relacionada.
Condiciones previas	Tener la tarea de la que queremos conocer la información instanciada correctamente dentro de un proyecto.
Salidas	Recuadro en el que aparece toda la información EPF relacionada de la tarea: Tipo de tarea, usuarios adicionales, usuario responsable y productos de trabajo asociados.
Post condición si éxito	Mostrar la información pedida al usuario.
Post condición si fallo	No se muestra la información.
Activador	Cualquier enlace que lleve a la pantalla de información de la tarea. Generalmente basta con hacer click sobre el nombre de la tarea en cualquier página de la aplicación donde ésta aparezca. Si no, siempre en el módulo "Tasks" de dotProject podremos acceder a todas las tareas existentes en el sistema. También podemos acceder a la información del proyecto (módulo Projects) al que pertenece y buscar la tarea entre todas las que conforman el proyecto.
Secuencia	1. Hacer click sobre el nombre de la tarea en cualquier punto indicado en la sección "Activador". Si la tarea tiene información EPF ya creada, ir a 2A. Si no, ir a 2B. 2A. En este punto estaremos en la página de información de la tarea. Podremos apreciar por el centro de la página que hay un recuadro amarillo. Ahí estará contenida la información EPF relacionada con la tarea. 2B. Al entrar en la página de información de tarea, apreciaremos que sobre el centro de la página aparecerá un recuadro amarillo con el texto "Add EPF Task info". Esto indica que la tarea aún no tiene información EPF asociada. Si se desea crear, mirar el caso de uso 1.3.
Excepciones	-
Frecuencia de uso	Media/Baja, ya que la visualización de la información EPF es algo importante pero no de uso excesivo.

Caso de uso 2.2	Visualización de roles
----------------------------	------------------------

Nivel de abstracción	2 Caso de uso padre: 2 – Ejecución de un proyecto FVE.
Objetivos	Lograr obtener información relacionada con los roles EPF dentro de nuestro sistema.
Actores	Cliente vía web, sistema.
Entradas	Proyecto/usuario/roles relacionados con la información que deseamos obtener.
Condiciones previas	Tener instanciado correctamente un proyecto para poder realizar a partir de él la recuperación de la información.
Salidas	La información relacionada con los roles demandada.
Post condición si éxito	Se muestra al usuario la información requerida relacionada con los roles.
Post condición si fallo	No se muestra ninguna información.
Activador	Diferentes enlaces repartidos por las páginas del proyecto.
Frecuencia de uso	Alta: los roles son un elemento muy importante dentro de FVE así como su ejecución. La visualización de la información relacionada con los roles será común a lo largo del proceso de desarrollo de un proyecto.

Caso de uso 2.2.1	Visualizar los roles que intervienen en un proyecto FVE
Nivel de abstracción	3 Caso de uso padre: 2.2 – Visualización de roles
Objetivos	Ver qué roles hay implementados dentro de un proyecto FVE y qué usuarios los implementan.
Actores	Cliente vía web, sistema.
Entradas	Proyecto del que deseamos conocer su información de roles.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema.
Salidas	Tabla en la que se listan todos los roles que aparecen en el proyecto junto a su descripción y los usuarios que los desempeñan.
Post condición si éxito	Información de roles demandada por el usuario.
Post condición si fallo	No se suministra ninguna información.
Activador	Pestaña “EPF Roles” en la página de información de proyecto.

Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de información del proyecto haciendo click en el nombre del proyecto en el módulo "Projects" de dotProject. 2. Una vez dentro de la página de información de proyecto, en la parte inferior se listarán en diversas pestañas los elementos que contiene este proyecto. Hacer click en la pestaña "EPF Roles". 3. Se mostrará el contenido de la pestaña, donde podremos ver todos los roles incluidos dentro de este proyecto junto a los usuarios que los implementan.
Excepciones	-
Frecuencia de uso	Baja/Media: Visualizar los roles existente en un proyecto puede ser de utilidad, aunque no por ello debe ser accedido con excesiva frecuencia.

Caso de uso 2.2.2	Visualizar los roles que intervienen en cualquier proyecto FVE del sistema
Nivel de abstracción	3 Caso de uso padre: 2.2 – Visualización de roles
Objetivos	Ver todos los roles pertenecientes a todos los proyectos creados en el sistema.
Actores	Cliente vía web, sistema.
Entradas	Todos los proyectos existentes en el sistema.
Condiciones previas	Todos los proyectos deben estar instanciados correctamente en el sistema.
Salidas	Informe con todos los roles existentes en cualquier proyecto del sistema, en el que se incluye además la descripción del rol y los usuarios que implementan ese rol.
Post condición si éxito	Generación por pantalla del informe con los datos de los roles.
Post condición si fallo	No se genera ninguna información.
Activador	Enlace "Roles" dentro del módulo FveProjectManager.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al módulo FveProjectManager. Para ello, éste debe estar habilitado y visible en las propiedades de los módulos en la configuración de dotProject. 2. Una vez dentro del módulo FveProjectManager, pulsar el enlace "Roles" de la parte superior de la página. 3. Se mostrará por pantalla la tabla con la información relacionada de los roles.
Excepciones	-
Frecuencia de uso	Baja: visualizar todos los roles insertados en el sistema puede ser útil para tener una visión general del sistema, pero lo más frecuente

	es comprobar los roles que pertenecen a un único proyecto.
--	--

Caso de uso 2.2.3	Ver información detallada de un rol EPF
Nivel de abstracción	3 Caso de uso padre: 2.2 – Ejecución de roles
Objetivos	Obtener la información específica de un rol EPF.
Actores	Cliente vía web, sistema.
Entradas	Rol del que queremos conocer la información detallada.
Condiciones previas	El rol ha de estar correctamente instanciado dentro de un proyecto del sistema.
Salidas	Página con toda la información específica del rol: <ul style="list-style-type: none"> • Información general: nombre, descripción y proyecto al que pertenece. • Relaciones con productos de trabajo: en que productos de trabajo interviene. • Relaciones con tareas: en qué tareas participa. • Usuario que implementan este rol.
Post condición si éxito	Se muestra por pantalla la página con la información específica del rol.
Post condición si fallo	No se muestra la información del rol.
Activador	Para acceder a la información del rol, hay que pulsar sobre el nombre del rol donde éste aparezca en cualquier punto de la aplicación. Si vamos a la pestaña “EPF Roles” de la página de información de un proyecto, podremos listar todos los roles y pinchar sobre el nombre del rol del que queramos conocer su información específica.
Secuencia	1. Mediante los activadores descritos en el anterior punto, accedemos a la página de información del rol.
Frecuencia de uso	Muy alta: revisar la información del rol será algo que tengamos que realizar periódicamente para consultar los datos necesarios durante el proceso de ejecución de nuestro proyecto.

Caso de uso 2.2.4	Visualizar todos los roles que puede desempeñar un usuario en los proyectos FVE
Nivel de abstracción	3 Caso de uso padre: 2.2 – Visualización de roles
Objetivos	Obtener un informe con los roles que es capaz de desarrollar un usuario dentro de todos los proyectos FVE en los que trabaja. Esto no

	significa que el usuario desarrolle esos roles, simplemente que es capaz de realizarlos.
Actores	Cliente vía web, sistema.
Entradas	Usuario del que queremos conocer qué roles puede desempeñar.
Condiciones previas	El usuario ha de estar correctamente creado.
Salidas	Página que contiene el informe con los roles que puede desempeñar el usuario, con la siguiente información: <ul style="list-style-type: none"> • Proyecto al que pertenece el rol • Nombre y descripción del rol
Post condición si éxito	Se muestra por pantalla la página con la información específica de los roles que puede desempeñar el usuario.
Post condición si fallo	No se muestra la información de los roles.
Activador	Pestaña "EPF Possible Roles" dentro la pantalla de información del usuario.
Secuencia	<p>1. Debemos acceder a la pantalla de información del usuario. Para ello, podemos o bien ir al módulo "User Admin" (debemos ser administradores para ello) o bien hacer click sobre el nombre del usuario en cualquier punto de la aplicación donde este aparezca (tareas, proyectos). Ir a 2.</p> <p>2. Estamos en la pantalla de información extendida del usuario. En la parte inferior hay varias pestañas que nos permiten obtener diversa información del usuario elegido. Hacer click en la pestaña "EPF Possible Roles". Pasar al punto 3.</p> <p>3. Tras hacer click en esta pestaña, se mostrará en el contenido de la misma en el marco inferior, donde obtendremos el informe con los roles que el usuario puede desempeñar y en qué proyectos.</p>
Frecuencia de uso	Media: es una información que puede ser de utilidad para valorar qué usuarios podemos asignar a las tareas o productos de trabajo que necesiten de un determinado rol.

Caso de uso 2.2.5	Visualizar todos los roles que desempeña un usuario dentro de los proyectos FVE en los que participa
Nivel de abstracción	3 Caso de uso padre: 2.2 – Visualización de roles
Objetivos	Obtener un informe con los roles que es actualmente está desarrollando un usuario dentro de todos los proyectos FVE en los que trabaja.
Actores	Cliente vía web, sistema.
Entradas	Usuario del que queremos conocer qué roles desempeña actualmente.

Condiciones previas	El usuario ha de estar correctamente creado.
Salidas	Página que contiene el informe con los roles que desempeña el usuario, con la siguiente información: <ul style="list-style-type: none"> • Proyecto al que pertenece el rol • Nombre y descripción del rol
Post condición si éxito	Se muestra por pantalla la página con la información específica de los roles que desempeña el usuario.
Post condición si fallo	No se muestra la información de los roles.
Activador	Pestaña "EPF Specific Roles" dentro la pantalla de información del usuario.
Secuencia	<p>1. Debemos acceder a la pantalla de información del usuario. Para ello, podemos o bien ir al módulo "User Admin" (debemos ser administradores para ello) o bien hacer click sobre el nombre del usuario en cualquier punto de la aplicación donde este aparezca (tareas, proyectos). Ir a 2.</p> <p>2. Estamos en la pantalla de información extendida del usuario. En la parte inferior hay varias pestañas que nos permiten obtener diversa información del usuario elegido. Hacer click en la pestaña "EPF Specific Roles". Pasar al punto 3.</p> <p>3. Tras hacer click en esta pestaña, se mostrará en el contenido de la misma en el marco inferior, donde obtendremos el informe con los roles que el usuario desempeña actualmente y en qué proyectos.</p>
Frecuencia de uso	Media: esta información ayuda a la hora de saber qué roles desempeña cada usuario y en qué proyectos para valorar así su capacidad de desarrollo y adaptabilidad.

Caso de uso 2.3	Ejecución de productos de trabajo
Nivel de abstracción	2 Caso de uso padre: 2 – Ejecución de un proyecto FVE.
Objetivos	Lograr obtener información relacionada con los productos EPF dentro de nuestro sistema.
Actores	Cliente vía web, sistema.
Entradas	Proyecto/usuario/productos de trabajo relacionados con la información que deseamos obtener.
Condiciones previas	Tener instanciado correctamente un proyecto para poder realizar a partir de él la recuperación de la información.
Salidas	La información relacionada con los productos de trabajo demandada o la ejecución de los propios productos de trabajo.
Post condición si	Se muestra al usuario la información requerida relacionada con los

éxito	productos de trabajo.
Post condición si fallo	No se muestra ninguna información.
Activador	Diferentes enlaces repartidos por las páginas del proyecto.
Frecuencia de uso	Alta: Los productos de trabajo son uno de los pilares fundamentales de EPF. Son los elementos que deben generar la tareas y los usuarios deben trabajar para su correcta terminación. La visualización y modificación del progreso de los mismos serán operaciones que se ejecuten frecuentemente durante el proceso de desarrollo del proyecto.

Caso de uso 2.3.1	Modificar el progreso de un producto de trabajo
Nivel de abstracción	3 Caso de uso padre: 2.3 – Ejecución de productos de trabajo
Objetivos	Modificar el porcentaje de terminación de un producto de trabajo.
Actores	Cliente vía web, sistema.
Entradas	Producto de trabajo del que queremos modificar su progreso.
Condiciones previas	Tener el producto de trabajo correctamente creado en el sistema y desempeñar como usuario un rol que sea responsable o modificador en el producto de trabajo a modificar.
Salidas	Mensaje de confirmación de que el estado del producto de trabajo ha sido correctamente modificado.
Post condición si éxito	El estado de terminación del producto de trabajo se modifica y se registra el usuario que lo realizó.
Post condición si fallo	El estado de terminación del producto de trabajo se mantiene igual.
Activador	Enlace "Edit status of Workproduct" en la pantalla de información del producto de trabajo.
Secuencia	<p>1. Acceder a la pantalla de información del producto de trabajo haciendo click en cualquier punto de la aplicación donde aparezca el producto de trabajo. Es posible igualmente ir a la pestaña "Workproducts" dentro de la pantalla de información del proyecto y seleccionar el producto de trabajo del que se quiera modificar su progreso. Ir al paso 2.</p> <p>2. Una vez dentro de la página de información del producto de trabajo, hacer click sobre el enlace "Edit status of Workproduct". Pasar a 3.</p> <p>3. Si el usuario no es ni usuario responsable ni modificador del producto de trabajo, ir a E-1. Si tiene algún puesto como responsable o modificador dentro del producto de trabajo, entonces se mostrará una barra deslizable para elegir el progreso que se</p>

	<p>desea establecer en el producto de trabajo. También se puede introducir manualmente el progreso en el campo "Status". Una vez hayamos introducido el valor que deseamos pulsamos el botón "Modify status" (ir a 4). Si deseamos cancelar la operación, debemos pulsar el botón "Back" (ir a 5).</p> <p>4. Tras haber confirmado la modificación del estado del producto de trabajo, volveremos a la página de información de éste, donde podremos comprobar que en la sección "Workproduct Status" aparecerá recogida esta modificación junto al nombre del último usuario que la realizó (en este caso, será su nombre de usuario el que aparezca).</p> <p>5. Volveremos de nuevo a la pantalla de información del producto de trabajo sin haber realizado ninguna modificación sobre éste.</p>
Excepciones	-
Frecuencia de uso	Muy Alta: esta modificación sobre los productos de trabajo se realizará de forma frecuente a medida que los usuarios lo vayan completando.

Caso de uso 2.3.2	Visualizar los productos de trabajo incluidos dentro de un proyecto FVE
Nivel de abstracción	3 Caso de uso padre: 2.3 – Ejecución de productos de trabajo
Objetivos	Ver qué productos de trabajo hay implementados dentro de un proyecto FVE y qué usuarios los implementan.
Actores	Cliente vía web, sistema.
Entradas	Proyecto del que deseamos conocer su información de productos de trabajo.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema.
Salidas	Tabla en la que se listan todos los productos de trabajo que aparecen en el proyecto junto a su descripción y su estado de terminación.
Post condición si éxito	Información de productos de trabajo demandada por el usuario.
Post condición si fallo	No se suministra ninguna información.
Activador	Pestaña "Workproduct" en la página de información de proyecto.
Secuencia	<p>1. Acceder a la página de información del proyecto haciendo click en el nombre del proyecto en el módulo "Projects" de dotProject.</p> <p>2. Una vez dentro de la página de información de proyecto, en la parte inferior se listarán en diversas pestañas los elementos que contiene este proyecto. Hacer click en la pestaña "Workproduct".</p> <p>3. Se mostrará el contenido de la pestaña, donde podremos ver todos los productos de trabajo incluidos dentro de este proyecto</p>

	junto a su estado de terminación.
Excepciones	-
Frecuencia de uso	Alta: este tipo de acceso a los productos de trabajo es el más común y se realizará frecuentemente durante la ejecución del proyecto.

Caso de uso 2.3.3	Visualizar todos los productos de trabajo incluidos en cualquier proyecto FVE
Nivel de abstracción	3 Caso de uso padre: 2.3 – Ejecución de productos de trabajo
Objetivos	Ver todos los productos de trabajo pertenecientes a todos los proyectos creados en el sistema.
Actores	Cliente vía web, sistema.
Entradas	Todos los proyectos existentes en el sistema.
Condiciones previas	Todos los proyectos deben estar instanciados correctamente en el sistema.
Salidas	Informe con todos los productos de trabajo existentes en cualquier proyecto del sistema, en el que se incluye además la descripción del producto de trabajo y el estado de terminación del mismo.
Post condición si éxito	Generación por pantalla del informe con los datos de los productos de trabajo.
Post condición si fallo	No se genera ninguna información.
Activador	Enlace "Workproducts" dentro del módulo FveProjectManager.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al módulo FveProjectManager. Para ello, éste debe estar habilitado y visible en las propiedades de los módulos en la configuración de dotProject. 2. Una vez dentro del módulo FveProjectManager, pulsar el enlace "Workproducts" de la parte superior de la página. 3. Se mostrará por pantalla la tabla con la información relacionada de los productos de trabajo.
Excepciones	-
Frecuencia de uso	Baja: visualizar todos los productos de trabajo insertados en el sistema puede ser útil para tener una visión general del sistema, pero lo más frecuente es comprobar los que ya pertenecen a un único proyecto.

Caso de uso 2.3.4	Ver información detallada de un producto de trabajo.
--------------------------	--

Nivel de abstracción	3 Caso de uso padre: 2.3 – Ejecución de productos de trabajo
Objetivos	Obtener la información específica de un producto de trabajo EPF.
Actores	Cliente vía web, sistema.
Entradas	Producto de trabajo del que queremos conocer la información detallada.
Condiciones previas	El producto de trabajo ha de estar correctamente instanciado dentro de un proyecto del sistema.
Salidas	Página con toda la información específica del producto de trabajo: <ul style="list-style-type: none"> • Información general: nombre, descripción y proyecto al que pertenece. • Plantilla asociada. • Relaciones con los roles y usuarios: qué rol es el responsable y cuáles los modificadores. Qué usuarios están asociados a dichos roles. • Relaciones con tareas: qué tareas poseen este producto de trabajo y con qué tipo de relación (salida, entrada obligatoria o entrada opcional). • Estado de terminación del producto de trabajo: qué porcentaje de terminación tiene el producto de trabajo y quién fue el último usuario que lo modificó.
Post condición si éxito	Se muestra por pantalla la página con la información específica del producto de trabajo.
Post condición si fallo	No se muestra la información del producto de trabajo.
Activador	Para acceder a la información del producto de trabajo, hay que pulsar sobre el nombre del producto de trabajo donde éste aparezca en cualquier punto de la aplicación. Si vamos a la pestaña "Workproducts" de la página de información de un proyecto, podremos listar todos los productos de trabajo y pinchar sobre el nombre del producto de trabajo del que queramos conocer su información específica.
Secuencia	1. Mediante los activadores descritos en el anterior punto, accedemos a la página de información del producto de trabajo.
Frecuencia de uso	Muy alta: revisar la información del producto de trabajo será algo que tengamos que realizar periódicamente para consultar los datos necesarios durante el proceso de ejecución de nuestro proyecto.

Caso de uso 2.3.4	Visualizar todos los productos de trabajo en los que participa un usuario
Nivel de abstracción	3 Caso de uso padre: 2.3 – Ejecución de productos de trabajo

Objetivos	Obtener todos los productos de trabajo en los que interviene un usuario.
Actores	Cliente vía web, sistema.
Entradas	Usuario del que queremos conocer los productos de trabajo en los que interviene.
Condiciones previas	El usuario debe estar correctamente definido dentro del sistema.
Salidas	Informe que contiene los productos de trabajo en los que interviene el usuario, organizado por proyectos. Para cada producto de trabajo se muestra: <ul style="list-style-type: none"> • Nombre del producto de trabajo. • Rol con el que el usuario participa en el producto de trabajo • Tipo de relación con el producto de trabajo: modificador o responsable.
Post condición si éxito	Se muestra por pantalla la página con los productos de trabajo en los que participa el usuario.
Post condición si fallo	No se muestra la información demandada.
Activador	Pestaña "EPF Workproducts" dentro de la pantalla de información del usuario.
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la pantalla de información del usuario del que queramos conocer en qué productos de trabajo interviene. Para ello, pulsar sobre su nombre en cualquier punto de la aplicación donde éste intervenga. 2. En la pantalla de información de usuario, pulsar sobre la pestaña "EPF Workproducts". 3. Se mostrará en el contenido de la pestaña de la parte inferior el informe con los casos de uso en los que participa el usuario.
Frecuencia de uso	Media: este caso de uso tiene una aplicación bastante específica pero que permite obtener una información que puede ser muy útil de carga a valorar la carga de trabajo que tiene un usuario.

Caso de uso 2.4	Instanciación de plantillas
Nivel de abstracción	2 Caso de uso padre: 2 – Ejecución de un proyecto FVE.
Objetivos	Lograr obtener información relacionada con las plantillas EPF dentro de nuestro sistema.
Actores	Cliente vía web, sistema.
Entradas	Proyecto/productos de trabajo/plantillas relacionados con la

	información que deseamos obtener.
Condiciones previas	Tener instanciado correctamente un proyecto para poder realizar a partir de él la recuperación de la información.
Salidas	La información relacionada con las plantillas demandadas.
Post condición si éxito	Se muestra al usuario la información requerida relacionada con las plantillas de trabajo.
Post condición si fallo	No se muestra ninguna información.
Activador	Diferentes enlaces repartidos por las páginas del proyecto.
Frecuencia de uso	Media: las plantillas son los archivos físicos asociados a los productos de trabajo. Como tal, son ejecutadas cuando el usuario ha realizado cambios sobre las mismas.

Caso de uso 2.4.1	Visualizar todas las plantillas asociadas a los productos de trabajo de un proyecto EPF
Nivel de abstracción	3 Caso de uso padre: 2.4 – Ejecución de plantillas
Objetivos	Ver qué plantillas hay implementados dentro de un proyecto FVE y en qué producto de trabajo están contenidas.
Actores	Cliente vía web, sistema.
Entradas	Proyecto del que deseamos conocer su información de sus plantillas.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema.
Salidas	Tabla en la que se listan todas las plantillas que aparecen en el proyecto junto a su descripción y los productos de trabajo a los que pertenecen.
Post condición si éxito	Información de plantillas demandada por el usuario.
Post condición si fallo	No se suministra ninguna información.
Activador	Pestaña "Templates" en la página de información de proyecto.
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de información del proyecto haciendo click en el nombre del proyecto en el módulo "Projects" de dotProject. 2. Una vez dentro de la página de información de proyecto, en la parte inferior se listarán en diversas pestañas los elementos que contiene este proyecto. Hacer click en la pestaña "Templates". 3. Se mostrará el contenido de la pestaña, donde podremos ver todas las plantillas incluidas dentro de este proyecto junto al

	producto de trabajo que la contiene.
Excepciones	-
Frecuencia de uso	Media/Alta: la visualización de la plantilla a partir de este método es muy útil porque permite ver de una vez todas las plantillas que existen en el proyecto. El método alternativo de visualizar las plantillas es a través del producto de trabajo que las contiene.

Caso de uso 2.4.2	Visualizar todas las plantillas de cualquier proyecto EPF
Nivel de abstracción	3 Caso de uso padre: 2.4 – Ejecución de plantillas
Objetivos	Ver todas las plantillas pertenecientes a todos los proyectos creados en el sistema.
Actores	Cliente vía web, sistema.
Entradas	Todos los proyectos existentes en el sistema.
Condiciones previas	Todos los proyectos deben estar instanciados correctamente en el sistema.
Salidas	Informe con todas las plantillas existentes en cualquier proyecto del sistema, en el que se incluye además la descripción de la plantilla y el producto de trabajo al que pertenece.
Post condición si éxito	Generación por pantalla del informe con los datos de las plantillas.
Post condición si fallo	No se genera ninguna información.
Activador	Enlace "Templates" dentro del módulo FveProjectManager.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al módulo FveProjectManager. Para ello, éste debe estar habilitado y visible en las propiedades de los módulos en la configuración de dotProject. 2. Una vez dentro del módulo FveProjectManager, pulsar el enlace "Templates" de la parte superior de la página. 3. Se mostrará por pantalla la tabla con la información relacionada de las plantillas.
Excepciones	-
Frecuencia de uso	Baja: visualizar todas las plantillas insertadas en los productos de trabajo del sistema puede ser útil para tener una visión general, pero lo más frecuente es comprobar las plantillas asociadas a un producto de trabajo o un proyecto concreto.

Caso de uso	Ver información detallada de una plantilla
--------------------	--

2.4.3	
Nivel de abstracción	3 Caso de uso padre: 2.4 – Ejecución de plantillas
Objetivos	Obtener la información específica de una plantilla EPF.
Actores	Cliente vía web, sistema.
Entradas	Plantilla de la que queremos conocer la información detallada.
Condiciones previas	La plantilla ha de estar correctamente instanciada dentro de un proyecto del sistema.
Salidas	Página con toda la información específica de la plantilla: <ul style="list-style-type: none"> • Información general: nombre, descripción y proyecto al que pertenece. • Producto de trabajo al que pertenece. • Archivo asociado a la plantilla, permitiendo acceder a él.
Post condición si éxito	Se muestra por pantalla la página con la información específica de la plantilla.
Post condición si fallo	No se muestra la información de la plantilla.
Activador	Para acceder a la información de la plantilla, hay que pulsar sobre el nombre de la plantilla donde ésta aparezca en cualquier punto de la aplicación. Si vamos a la pestaña "Templates" de la página de información de un proyecto, podremos listar todas las plantillas y pinchar sobre el nombre de la plantilla de la que queramos conocer su información específica.
Secuencia	1. Mediante los activadores descritos en el anterior punto, accedemos a la página de información de la plantilla.
Frecuencia de uso	Alta: revisar la información de la plantilla será algo que tengamos que realizar periódicamente para consultar los datos necesarios durante el proceso de ejecución de nuestro proyecto.

Caso de uso 2.4.4	Descarga del archivo asociado a la plantilla
Nivel de abstracción	3 Caso de uso padre: 2.4 – Ejecución de plantillas
Objetivos	Descargar a nuestro sistema de archivos el fichero que esté asociado a una plantilla.
Actores	Cliente vía web, sistema.
Entradas	La plantilla de la cual queremos descargar el archivo.
Condiciones	La plantilla debe tener asociado correctamente un archivo.

previas	
Salidas	El archivo físico. Se mostrará por el propio navegador web o se consultará al usuario por si desea descargarlo a su sistema de ficheros.
Post condición si éxito	Se descarga al sistema de ficheros del usuario el archivo asociado a la plantilla.
Post condición si fallo	No se consigue descargar el archivo.
Activador	Enlace "Press here to link to the file or right click to download it" dentro de la ventana de información de la plantilla. Icono con forma de hoja situado al lado del nombre de la plantilla dentro de la pantalla de información específica de un producto de trabajo.
Secuencia	<p>1A. Debemos acceder a la página de información del producto de trabajo que contiene la plantilla de la que deseamos recuperar. Para ello pulsamos sobre el nombre del producto de trabajo en cualquier punto de la aplicación donde éste aparezca. Ir a 2A.</p> <p>2A. Una vez estemos en la pantalla de información del producto de trabajo, en el apartado "Template associated" se listará una plantilla si esta existe. Podemos pulsar sobre su nombre e ir a 2B o pulsar sobre el icono con forma de folio e ir a 3.</p> <p>1B. Accedemos a la página de información de la plantilla de la que queremos recuperar su archivo. Ir a 2B.</p> <p>2B. En la página de información de la plantilla, si existe un archivo asociado encontraremos un enlace que pone "Press here to link to the file or right click to download it". Pulsarlo e ir a 3.</p> <p>3. Si el archivo es tratable por el navegador web, ir a 4. Si no, ir a 5.</p> <p>4. El archivo puede ser interpretado por el navegador (como por ejemplo una imagen o un texto) así que se mostrará en el mismo espacio de navegación del navegador.</p> <p>5. De no ser posible interpretar el archivo, aparecerá un explorador de archivos para que el usuario pueda elegir la ruta donde desea guardar el fichero.</p>
Excepciones	E-1: Si el archivo no se puede encontrar en el servidor, aparecerá un mensaje de error de "Objeto no encontrado".
Frecuencia de uso	Alta: la descarga de los archivos asociados a las plantillas será una operación frecuente que realizemos en el sistema, ya que es la principal vía de conocer el contenido de este archivo.

Caso de uso 2.5	Ejecución del repositorio de Subversion
Nivel de abstracción	2 Caso de uso padre: 3 – Ejecución de un proyecto FVE.
Objetivos	Llevar a cabo operaciones con nuestro repositorio subversion

	asociado al un proyecto en cuestión.
Actores	Cliente vía web, sistema.
Entradas	Repositorio de subversion con el que queremos trabajar.
Condiciones previas	El repositorio debe estar correctamente configurado y en funcionamiento.
Salidas	Confirmación de que la interacción con el repositorio se ha producido con éxito.
Post condición si éxito	El usuario logra sus objetivos de cara al repositorio.
Post condición si fallo	El usuario no logra lo que buscaba al ejecutar el repositorio.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Alta: la interacción con el repositorio es uno de los pilares de la aplicación y será muy usado por los usuarios durante la ejecución de sus proyectos.

Caso de uso 2.5.1	Acceder al panel de gestión de subversion asociado a un proyecto
Nivel de abstracción	3 Caso de uso padre: 2.5 – Ejecución del repositorio de Subversion
Objetivos	Acceder al menú de subversion del proyecto para poder trabajar sobre él.
Actores	Cliente vía web, sistema.
Entradas	Pestaña
Condiciones previas	El proyecto ha de ser instanciado
Salidas	<p>En la parte inferior de la página de visualización de un proyecto se sustituye la pestaña residente por una compuesta por dos paneles:</p> <ol style="list-style-type: none"> 1. El primero está situado a la izquierda y muestra la representación de un árbol jerárquico de exploración del repositorio de subversion asociado con el proyecto. 2. El segundo contiene las distintas opciones de trabajo que la aplicación ofrece sobre subversion: <ol style="list-style-type: none"> 2.1. Obtener la información del repositorio de subversion asociado al proyecto. 2.2. Descargar bajo formato zip una copia de trabajo del repositorio asociado al proyecto. 2.3. Acceder al gestor de archivos "FileManager" que ofrece la

	<p>aplicación sobre el repositorio asociado al proyecto</p> <p>2.4. Realizar una importación al repositorio asociado al proyecto mediante la subida de un fichero zip con los contenidos a importar.</p> <p>2.5. Realizar una importación al repositorio asociado al proyecto mediante la subida de un fichero zip con los contenidos a importar.</p>
Post condición si éxito	Se muestra por pantalla en la parte inferior de la página la pestaña de subversion descrita en Salidas.
Post condición si fallo	-
Activador	Para acceder a este menú ha de seleccionarse la pestaña marcada como "Subversion" en la parte inferior de la página de información de un proyecto.
Secuencia	<ol style="list-style-type: none"> 1. Accedemos a la ventana de listado de proyectos 2. Elegimos el proyecto a visualizar 3. En la parte inferior de la pantalla pulsamos la pestaña de "Subversion" 4. Visualización en la parte inferior de la pantalla descrita en Salidas. <p>E1: En caso de no estar el módulo activo</p>
Excepciones	E1
Frecuencia de uso	Alta: Cada vez que deseamos trabajar con el módulo de subversion hemos de acceder a este menú.

Caso de uso 2.5.1.1	Visualización de la información del repositorio de un proyecto
Nivel de abstracción	4 Caso de uso padre: 2.5.1 – Acceder al panel de gestión de subversion asociado a un proyecto.
Objetivos	Establecer la disponibilidad del sistema del módulo anexo de JSP que permite realizar las operaciones sobre subversion
Actores	Cliente vía web, sistema.
Entradas	Botón de acceso
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Modificación de la propiedad en la ventana del panel de configuración de subversion.
Post condición si éxito	Visualización de la página de información asociada al repositorio del proyecto.
Post condición si	-

fallo	
Activador	Botón de "Information" situado en el panel de la derecha del menú de subversion.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Una vez dentro del menú, pulsar el botón de "Information" situado en la parte superior del panel de la derecha. 3. Visualización de la información del repositorio a través de una página. E1:
Excepciones	E1: En caso de no encontrar dicha información se muestra la página destino con el mensaje "This Project doesn't have that kind of information."
Frecuencia de uso	Baja: la información del repositorio no es dato que se consulte habitualmente.

Caso de uso 2.5.1.2	Obtención de la copia de trabajo del repositorio asociado a un proyecto
Nivel de abstracción	4 Caso de uso padre: 2.5.1 – Acceder al panel de gestión de subversion asociado a un proyecto.
Objetivos	Disponer de un archivo comprimido con los contenidos de una copia de trabajo asociada al repositorio del proyecto instanciado.
Actores	Cliente vía web, sistema.
Entradas	Botón de acceso
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Obtención de un archivo zip con los contenidos descritos anteriormente.
Post condición si éxito	-
Post condición si fallo	-
Activador	Botón de "Checkout" situado en el panel de la derecha del menú de subversion.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Una vez dentro del menú, pulsar el botón de "Checkout" situado en la parte superior del panel de la derecha. 3. Emerge una ventana con la opciones de descarga del archivo. E1 4. Una vez descargado podemos trabajar con su contenido.
Excepciones	E1: Si el repositorio del proyecto es demasiado grande, el explorador puede no responder o colapsarse.

Frecuencia de uso	Media: Los usuarios necesitan acceder con relativa frecuencia a esta opción para descargarse de una manera sencilla los archivos del repositorio actualizados.
-------------------	--

Caso de uso 2.5.1.3	Despliegue y visualización del contenido del repositorio a través de su árbol de exploración
Nivel de abstracción	4 Caso de uso padre: 2.5.1 – Acceder al panel de gestión de subversion asociado a un proyecto.
Objetivos	Actualizar los contenidos (archivos existentes y no existentes) del repositorio mediante un archivo zip con los nuevos contenidos.
Actores	Cliente vía web, sistema.
Entradas	
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2) o residir en el "FileManager" de subversion.
Salidas	Según la opción pulsada en el árbol: <ol style="list-style-type: none"> Si pulsamos sobre una cruz de despliegue de carpetas: <ul style="list-style-type: none"> -El árbol se contraerá/desplegará en ese punto para mostrar/ocultar el contenido de la carpeta asociada. Si pulsamos sobre el icono de una carpeta o su nombre: <ul style="list-style-type: none"> -Accederemos a la página del gestor de archivos "FileManager" que la aplicación ofrece , con la dirección del directorio activo el que hemos pulsado. Si pulsamos sobre el icono de un fichero o su nombre: <ul style="list-style-type: none"> -El explorador abrirá el contenido del fichero si su extensión se lo permite, o bien le ofrecerá descargarse el archivo.
Post condición si éxito	-
Post condición si fallo	-
Activador	Objetos del árbol de carpetas del repositorio asociado al proyecto.
Secuencia	<ol style="list-style-type: none"> Acceder al menú de subversion o al gestor de archivos "FileManager" Pulsar los elementos del árbol que despliegan los directorios para visualizar el archivo o carpeta deseados. Si pulsamos sobre un fichero ir a 4. Si pulsamos sobre un directorio ir a 5. La página es redirigida al contenido del fichero: <ol style="list-style-type: none"> Si el fichero tiene una extensión conocida se abrirá en la ventana y fin. Sino ir a 4.2 La ventana no cambiará y emergerá el panel de descarga del

	fichero. 5. Se abre la ventana del gestor de archivos "FileManager" de Subversion en la ubicación seleccionada(mostrando el contenido del directorio seleccionado en el árbol)
Frecuencia de uso	Muy Alta: Navegar a través del árbol del repositorio para acceder a los contenidos es una acción que se realiza varias veces en cada sesión.

Caso de uso 2.5.1.4	Acceder al gestor de archivos de subversion "File Manager"
Nivel de abstracción	4 Caso de uso padre: 2.5.1 – Acceder al panel de gestión de subversion asociado a un proyecto.
Objetivos	Mostrar una ventana que visualice el contenido de una carpeta del repositorio en la cual estemos "situados" y las diversas operaciones de inserción, modificación y eliminación de archivos que se pueden realizar.
Actores	Cliente vía web, sistema.
Entradas	Botón en el panel de gestión de subversion(o mediante navegación del árbol de exploración)
Condiciones previas	El proyecto ha de ser instanciado, el módulo de subversion ha de estar configurado como activo
Salidas	Se visualiza una página nueva con el siguiente contenido: <ul style="list-style-type: none"> • Una franja superior con el nombre del proyecto y la ruta relativa en la que está situado el gestor dentro del repositorio • Una franja con un formulario y botones que permiten subir y crear archivos y carpetas en el repositorio en la carpeta actual. • Debajo de una línea se muestran todos los archivos y carpetas que el directorio actual del repositorio contiene. • A la derecha un árbol de exploración del repositorio
Post condición si éxito	Se muestra por pantalla la ventana descrita en Salidas
Post condición si fallo	-
Activador	Para acceder a este menú ha de seleccionarse la pestaña marcada como "Subversion" en la parte inferior de la página de información de un proyecto.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Pulsamos el botón "FileManager" 3. Visualización ventana descrita en Salidas
Frecuencia de uso	Muy Alta: Cada vez que deseamos trabajar con el contenido de los ficheros de subversion hemos de acceder a este gestor.

Caso de uso 2.5.1.4.1	Navegación a través del repositorio mediante el uso del gestor de archivos(File Manager)
Nivel de abstracción	5 - Caso de uso padre: 2.5.1.4 – Acceder al gestor de archivos de subversion "File Manager"
Objetivos	Actualizar los contenidos(archivos existentes y no existentes) del repositorio mediante un archivo zip con los nuevos contenidos.
Actores	Cliente vía web, sistema.
Entradas	
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Según la opción pulsada en el árbol: <ol style="list-style-type: none"> Si pulsamos sobre la carpeta de : -El árbol se contraerá/desplegará en ese punto para mostrar/ocultar el contenido de la carpeta asociada. Si pulsamos sobre el icono de una carpeta o su nombre: -Accederemos a la página del gestor de archivos "FileManager" que la aplicación ofrece , con la dirección del directorio activo el que hemos pulsado. Si pulsamos sobre el icono de un fichero o su nombre: -El explorador abrirá el contenido del fichero si su extensión se lo permite, o bien le ofrecerá descargarse el archivo.
Post condición si éxito	-
Post condición si fallo	-
Activador	Objetos del árbol de carpetas del repositorio asociado al proyecto.
Secuencia	<ol style="list-style-type: none"> Acceder al menú de subversion o al gestor de archivos "FileManager" Pulsar los elementos del árbol que despliegan los directorios para visualizar el archivo o carpeta deseada. Si pulsamos sobre un fichero ir a 4.Si pulsamos sobre un directorio ir a 5. La página es redirigida al contenido del fichero: <ol style="list-style-type: none"> Si el fichero tiene una extensión conocida se abrirá en la ventana y fin. Sino ir a 4.2 Ventana no cambiará y emergerá el panel de descarga del fichero. Se abre la ventana del gestor de archivos "FileManager" de Subversion en la ubicación seleccionada(mostrando el contenido del directorio seleccionado en el árbol)
Frecuencia de uso	Muy Alta: Navegar a través del árbol del repositorio para acceder a los contenidos es una acción que se realiza varias veces en cada

	sesión.
--	---------

3.3.3 Modificación de un proyecto FVE

Caso de uso 3	Modificación de un proyecto FVE
Nivel de abstracción	1
Objetivos	Modificar los parámetros de un proyecto FVE que esté instanciado en el sistema.
Actores	Cliente vía web, sistema.
Entradas	Proyecto FVE instanciado en el sistema.
Condiciones previas	El proyecto debe estar correctamente definido y en uso.
Salidas	Confirmación de que la modificación del proyecto ha sido correcta.
Post condición si éxito	El proyecto queda modificado en alguno de sus aspectos: tareas, productos de trabajo, roles o plantillas.
Post condición si fallo	El proyecto no sufre modificación alguna.
Activador	-
Frecuencia de uso	Muy alta: la edición de los parámetros que conforman un proyecto será una operación que se realice con frecuencia, ya que el desarrollo de un proyecto es un proceso cambiante que requiere de continuas revisiones y ediciones para que se ajuste a las necesidades reales de ese proyecto.

Caso de uso 3.1	Modificación de parámetros de un proyecto y sus tareas
Nivel de abstracción	2 Caso de uso padre: 3 – Modificación de un proyecto FVE.
Objetivos	Modificar la información EPF asociada a un proyecto o las tareas que éste contiene.
Actores	Cliente vía web, sistema.
Entradas	El proyecto al que queremos realizar modificaciones.
Condiciones previas	El proyecto debe estar correctamente instanciado y tener información EPF asociado y al menos una tarea insertada (si queremos realizar su modificación).
Salidas	Confirmación de que la modificación ha tenido éxito.

Post condición si éxito	El proyecto con sus tareas o información EPF modificada.
Post condición si fallo	El proyecto no sufre modificación alguna.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Media: los parámetros EPF del proyecto rara vez sufrirán modificaciones si ya están configurados. Sin embargo, la edición de las tareas es una acción que se realizará con relativa frecuencia.

Caso de uso 3.1.1	Editar la información EPF de un proyecto
Nivel de abstracción	3 Caso de uso padre: 3.1 – Modificación de un proyecto FVE
Objetivos	Modificar los datos EPF asociados a un proyecto, como el enlace con la información del proyecto, el path del repositorio o los datos del plugin importado.
Actores	Cliente vía web, sistema.
Entradas	El proyecto del que queremos modificar las propiedades EPF.
Condiciones previas	El proyecto debe estar correctamente instanciado y debe ser un proyecto FVE, lo que implica que tenga propiedades EPF.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	Las propiedades EPF del proyecto son modificadas acorde a los cambios introducidos por el usuario.
Post condición si fallo	No se modifican las propiedades del proyecto.
Activador	Botón “EPF Project Info” en la pantalla de información general del proyecto.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto. Para ello, hay varios métodos. El más sencillo es acudir al módulo “Projects” de dotProject y pulsar sobre el proyecto del que queramos modificar sus propiedades EPF. Ir 2.</p> <p>2. Una vez en la pantalla de información del proyecto, debemos pulsar el botón “EPF Project Info” que se encuentra en el recuadro amarillo con las propiedades EPF del proyecto. Si el proyecto no tenía propiedades EPF anteriormente, mirar el caso de uso 1.2. Si ya tenía propiedades EPF asociadas, ir a 3.</p> <p>3. Tras pulsar el botón de edición, entraremos en la primera pantalla de edición de los parámetros EPF del proyecto. Podremos ver los</p>

	<p>valores que actualmente tiene asociado el proyecto en su configuración EPF. Modificar en los recuadros de texto los valores que creamos convenientes. Cuando estemos listos, pulsar el botón "Edit Project". Ir a 4.</p> <p>4. Ahora podemos editar los usuarios, pero este proceso está recogido en el caso de uso 3.1.2. Pulsamos el botón "Finish EPF Editing". Ir a 5.</p> <p>5. Tras haber realizado correctamente el proceso de edición de los parámetros EPF del proyecto, se mostrará una pantalla de confirmación en la que se indicará que la edición fue correcta. Pulsamos sobre el enlace "< Go back to project info page". Ir a 6.</p> <p>6. Tras haber realizado correctamente la edición del proyecto, volveremos a la pantalla de información del proyecto donde veremos patentes los cambios que hemos realizado sobre él.</p>
Excepciones	E-1: Si no se introduce una ruta de repositorio válida, se producirá un error.
Frecuencia de uso	Baja: una vez fijados los parámetros EPF de un proyecto en su creación, rara vez será necesario que éstos sean modificados.

Caso de uso 3.1.2	Editar los usuarios que pueden trabajar en un proyecto FVE
Nivel de abstracción	3 Caso de uso padre: 3.1 – Modificación de un proyecto FVE
Objetivos	Modificar los usuarios que podrán trabajar en un proyecto FVE.
Actores	Cliente vía web, sistema.
Entradas	El proyecto del que queremos modificar los usuarios que trabajan en él.
Condiciones previas	El proyecto debe estar correctamente instanciado y debe ser un proyecto FVE, lo que implica que tenga propiedades EPF y usuarios asociados a él.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	Los usuarios que trabajan en el proyecto son modificados acorde a los cambios introducidos por el usuario.
Post condición si fallo	No se modifican los usuarios del proyecto.
Activador	Botón "EPF Project Info" en la pantalla de información general del proyecto.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos modificar sus propiedades EPF. Ir 2.</p> <p>2. Una vez en la pantalla de información del proyecto, debemos pulsar el botón "EPF Project Info" que se encuentra en el recuadro</p>

	<p>amarillo con las propiedades EPF del proyecto. Si el proyecto no tenía propiedades EPF anteriormente, mirar el caso de uso 1.2. Si ya tenía propiedades EPF asociadas, ir a 3.</p> <p>3. Tras pulsar el botón de edición, entraremos en la primera pantalla de edición de los parámetros EPF del proyecto. Esta edición se recoge en el caso de uso 3.1.1. Cuando estemos listos, pulsar el botón "Edit Project". Ir a 4.</p> <p>4. Ahora podemos editar los usuarios. Para ello, elegimos los usuarios que deseamos que participen en el proyecto en la lista desplegable que contiene todos los usuarios del sistema y pulsamos "Add user" para añadirlo. Cuando estemos satisfechos con los usuarios insertados, pulsamos el botón "Finish EPF Editing". Ir a 5.</p> <p>5. Tras haber realizado correctamente el proceso de edición de los parámetros EPF y los usuarios del proyecto, se mostrará una pantalla de confirmación en la que se indicará que la edición fue correcta. Pulsamos sobre el enlace "< Go back to project info page". Ir a 6.</p> <p>6. Tras haber realizado correctamente la edición del proyecto, volveremos a la pantalla de información del proyecto donde veremos patentes los cambios que hemos realizado sobre él.</p>
Excepciones	-
Frecuencia de uso	Baja/Media: esta funcionalidad se puede emplear si deseamos realizar cambios sobre los usuarios insertados sobre el proyecto, como añadir un nuevo usuario que no existía en el momento de crear el proyecto o borrar a uno que ya no debe participar en el mismo.

Caso de uso 3.1.3	Editar la información EPF de una tarea
Nivel de abstracción	3 Caso de uso padre: 3.1 – Modificación de un proyecto FVE
Objetivos	Editar la información complementaria EPF sobre una tarea que se utiliza en nuestra aplicación FVE.
Actores	Cliente vía web, sistema.
Entradas	La tarea sobre la que queremos editar su información EPF.
Condiciones previas	La tarea debe de estar correctamente instanciada dentro de un proyecto FVE y debe tener asociada información EPF.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información EPF de la tarea quedará editada con los nuevos parámetros introducidos por el usuario.
Post condición si fallo	No se modifican los parámetros EPF de la tarea.
Activador	Botón "EPF Task Info" en la pantalla de información general de la tarea.

Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información de la tarea. Para ello, hay varios métodos. El más sencillo es acudir a la pestaña "Tasks" dentro del proyecto que albergue la tarea de la que queremos modificar sus propiedades. Ir 2.</p> <p>2. Una vez en la pantalla de información de la tarea, debemos pulsar el botón "EPF Project Info" que se encuentra en el recuadro amarillo con las propiedades EPF de la tarea. Si la tarea no tenía propiedades EPF anteriormente, mirar el caso de uso 1.2. Si ya tenía propiedades EPF asociadas, ir a 3.</p> <p>3. Tras pulsar el botón de edición, entraremos en la primera pantalla de edición de los parámetros EPF de la tarea. Podemos modificar el tipo de tarea sobrescribiéndolo en el recuadro "Task type". Cuando estemos listos, pulsar el botón "Next step". Ir a 4.</p> <p>4. Ahora podemos editar los roles y los productos de trabajo que intervienen en la tarea. En la parte superior, podemos elegir el rol que queremos que sea primario en la tarea y después tantos roles adicionales como queramos. En la sección de Productos de trabajo, modificamos los workproducts con el tipo de dependencia que deseemos. Si introducimos un producto de trabajo que ya exista con el mismo tipo de dependencia, ir a E-1. Cuando hayamos terminado de modificar estos elementos, pulsar "Next step" e ir a 5.</p> <p>5. Tras haber realizado correctamente el proceso de edición de los roles y los productos de trabajo de la tarea, podemos modificar los usuarios que realmente están asociados a los roles que configuramos en el paso 5. Seleccionamos el primario mediante la lista desplegable y añadimos y quitamos usuarios modificadores mediante los botones add user y remove user. Cuando hayamos modificado esta información, pulsamos el botón "Finish" y vamos al punto 6.</p> <p>6. Tras haber realizado correctamente el proceso de edición del tipo de tarea, roles, productos de trabajo y usuarios, se mostrará una pantalla de confirmación en la que se indicará que la edición fue correcta. Pulsamos sobre el enlace "< Go back to task info". Ir a 7.</p> <p>7. Tras haber realizado correctamente la edición de la tarea, volveremos a la pantalla de información de la tarea donde veremos patentes los cambios que hemos realizado sobre ella.</p>
Excepciones	E-1: Se mostrará por pantalla un mensaje informativo relativo a que no se puede insertar el mismo producto de trabajo con la misma dependencia con la tarea.
Frecuencia de uso	Media: esta funcionalidad se puede emplear si deseamos realizar cambios sobre los usuarios que trabajan en una tarea o los productos de trabajo asociados. Es una operación que puede ser necesaria de realizar durante la ejecución del proyecto.

Caso de uso 3.1.4	Borrar un proyecto FVE
Nivel de abstracción	3 Caso de uso padre: 3.1 – Modificación de un proyecto FVE
Objetivos	Borrar un proyecto FVE del sistema.

Actores	Cliente vía web, sistema.
Entradas	El proyecto que se desea borrar
Condiciones previas	El proyecto debe estar correctamente instanciado dentro del sistema.
Salidas	Confirmación de que el borrado se realizó correctamente.
Post condición si éxito	El proyecto FVE y todos sus elementos EPF (tareas, productos de trabajo, plantillas) quedan eliminados del sistema. También se borrará el repositorio SVN que tenga el proyecto asociado.
Post condición si fallo	El proyecto no se borra y permanece en el sistema.
Activador	Enlace "delete project" en la pantalla de información general del proyecto.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos modificar sus propiedades EPF. Ir 2.</p> <p>2. Una vez en la pantalla de información del proyecto, debemos pulsar el botón "delete Project" que se encuentra en la parte superior derecha del cuadro de información del proyecto. Ir a 3.</p> <p>3. Se mostrará una ventana desplegable que preguntará al usuario si desea borrar el proyecto. Si se pulsa el botón "aceptar" ir a 4, sino ("cancelar") ir a 5.</p> <p>4. Volveremos a la página que muestra todos los proyectos existentes en el sistema (donde ya no aparece el que acabamos de borrar) y en la parte superior de la pantalla aparecerá el mensaje "Project deleted" confirmando que el proyecto fue correctamente borrado.</p> <p>5. Hemos elegido no borrar el proyecto. Volveremos a la página de información general del proyecto y éste no habrá sido borrado.</p>
Excepciones	-
Frecuencia de uso	Baja: no es una operación frecuente borrar proyectos. Se realizará una vez estén completados o cuando no estemos contentos con su planteamiento general y deseemos instanciar uno nuevo.

Caso de uso 3.1.5	Borrar una tarea EPF
Nivel de abstracción	3 Caso de uso padre: 3.1 – Modificación de un proyecto FVE
Objetivos	Borrar una tarea EPF del sistema.

Actores	Cliente vía web, sistema.
Entradas	La tarea que se desea borrar.
Condiciones previas	La tarea debe estar correctamente instanciada en el sistema y pertenecer a un proyecto.
Salidas	Confirmación de que el borrado de la tarea se realizó correctamente.
Post condición si éxito	La tarea queda eliminada del sistema.
Post condición si fallo	No se borra la tarea.
Activador	Enlace "delete task" en la pantalla de información general de la tarea.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información de la tarea. Para ello, hay varios métodos. El más sencillo es acudir a la pestaña "Tasks" dentro del proyecto que albergue la tarea de la que queremos modificar sus propiedades. Ir 2.</p> <p>2. Una vez estemos en la pantalla con la información de la tarea, debemos pulsar el enlace "Delete Task" que se encuentra en la parte superior derecha del cuadro con la información de la tarea.</p> <p>3. Se mostrará una ventana desplegable que preguntará al usuario si desea borrar la tarea. Si se pulsa el botón "aceptar" ir a 4, sino ("cancelar") ir a 5.</p> <p>4. Volveremos a la página de información general del proyecto al que pertenecía la tarea (donde ya no aparece la que acabamos de borrar) y en la parte superior de la pantalla aparecerá el mensaje "Task deleted" confirmando que la tarea fue correctamente borrada.</p> <p>5. Hemos elegido no borrar la tarea. Volveremos a la página de información general de la tarea y ésta no habrá sido borrada.</p>
Excepciones	-
Frecuencia de uso	Media: borrar tareas puede ser una operación frecuente ya que la edición, creación y borrado de tareas serán operaciones comunes ya que las tareas son los elementos donde realmente se asignan los usuarios para trabajar.

Caso de uso 3.2	Modificación de roles de un proyecto
Nivel de abstracción	2 Caso de uso padre: 3 – Modificación de un proyecto FVE.
Objetivos	Realizar modificaciones sobre los roles EPF que intervienen dentro de un proyecto instanciado en el sistema.
Actores	Cliente vía web, sistema.

Entradas	El proyecto al que queremos realizar modificaciones sobre sus roles.
Condiciones previas	El proyecto debe estar correctamente instanciado y tener información EPF asociada.
Salidas	Confirmación de que la modificación ha tenido éxito.
Post condición si éxito	El proyecto con sus roles modificados.
Post condición si fallo	El proyecto no sufre modificación alguna.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Baja/Media: la modificación de los roles es algo relativamente utilizado. Los roles serán lo que permita asociar a los usuarios a las tareas y los productos de trabajo, por lo que su edición determinará en buena manera esa asociación entre usuarios y tareas o productos de trabajo.

Caso de uso 3.2.1	Borrar un rol EPF
Nivel de abstracción	3 Caso de uso padre: 3.2 – Modificación de roles de un proyecto
Objetivos	Borrar un rol EPF del sistema.
Actores	Cliente vía web, sistema.
Entradas	El rol EPF que se desea borrar.
Condiciones previas	La rol debe estar correctamente instanciada en el sistema y pertenecer a un proyecto. El rol no debe tener ningún usuario asociado dentro de un producto de trabajo o una tarea, ya que si se borrara así podría generar inconsistencias en el sistema.
Salidas	Confirmación de que el borrado del rol se realizó correctamente.
Post condición si éxito	El rol queda eliminado del sistema y todas sus asociaciones con los usuarios.
Post condición si fallo	No se borra el rol.
Activador	Icono 'papelera' junto al rol en las pantallas en las que se lista el rol, como por ejemplo en la pestaña "EPF Roles" en la pantalla de información de proyecto.
Secuencia	1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene el rol. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus roles. Ir 2.

	<p>2. En la pantalla de información del proyecto que contiene el rol que queremos borrar, pulsamos sobre la pestaña “EPF Roles”. Ir a 3.</p> <p>3. Se mostrará el listado con todos los roles que contiene el proyecto. Elegimos el rol que queramos borrar y hacemos click sobre el icono con forma de papelera que aparece en la columna “Operations”. Ir a 4.</p> <p>4. Se mostrará la pantalla de borrado del rol. Si el rol tiene usuarios asociados en productos de trabajo o tareas, ir a 5, sino ir a 6.</p> <p>5. Si el rol tiene usuarios asociados en productos de trabajo o tareas con este rol, no se podrá borrar. Para ello deberemos borrar o modificar previamente estas asociaciones en los detalles del rol. En esta página se nos informará de estas asociaciones para poder tomar medidas de cara a la eliminación del rol. Pulsar el botón “Don’t delete” e ir a 9.</p> <p>6. El rol está libre de asociaciones, por lo que se puede borrar sin que cree ninguna inconsistencia en el sistema. Pulsamos el botón “Delete” si queremos confirmar el borrado y vamos a 7. Si no queremos borrar finalmente el rol, pulsamos el botón “Don’t delete” e ir a 9.</p> <p>7. Tras borrar el rol se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace “Go back to Project Info”. Ir a 8.</p> <p>8. Tras haber borrado el rol, volveremos a la pantalla de información del proyecto que contenía el rol.</p> <p>9. Volveremos a la pantalla de información del proyecto que contenía el rol sin que éste haya sido borrado.</p>
Excepciones	-
Frecuencia de uso	Baja: el borrado de los roles necesita que éstos no tengan asociados usuarios que estén trabajando con ese rol en tareas o productos de trabajo. Por lo tanto, para borrar el rol debemos eliminar estas asociaciones previamente. Es una operación costosa, y además el borrado de roles no es una operación que se realice con asiduidad dentro de la ejecución de un proyecto.

Caso de uso 3.2.2	Editar un rol EPF
Nivel de abstracción	3 Caso de uso padre: 3.2 – Modificación de roles de un proyecto
Objetivos	Editar la información asociada a un rol EPF: su descripción y los usuarios que lo pueden desempeñar.
Actores	Cliente vía web, sistema.
Entradas	El rol EPF que queremos editar.
Condiciones previas	El rol EPF debe estar correctamente creado y pertenecer a un proyecto del sistema.

Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información asociada al rol EPF quedará modificada.
Post condición si fallo	No se modificará la información EPF del rol.
Activador	Icono 'libreta' junto al rol en las pantallas en las que se lista el rol, como por ejemplo en la pestaña "EPF Roles" en la pantalla de información de proyecto.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene el rol. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus roles. Ir 2.</p> <p>2. En la pantalla de información del proyecto que contiene el rol que queremos borrar, pulsamos sobre la pestaña "EPF Roles". Ir a 3.</p> <p>3. Se mostrará el listado con todos los roles que contiene el proyecto. Elegimos el rol que queramos borrar y hacemos click sobre el icono con forma de libreta que aparece en la columna "Operations". Ir a 4.</p> <p>4. Se mostrará la primera página de edición del rol. En esta podremos modificar la descripción del mismo en el campo "Brief description" cuando estemos listos, pulsar "Edit Role" e ir a 5, si queremos cancelar la edición pulsar "Back" e ir a 9.</p> <p>5. Tras editar la descripción del rol, pasaremos a la pantalla de configuración de usuarios. Podemos modificar qué usuarios participan en el rol añadiéndolos con el botón "Add user" y quitándolos con el botón "Remove User". Si queremos volver a la pantalla de edición de la descripción del rol, pulsar el botón "Back" e ir a 4. Si estamos contentos con la edición de la descripción y los usuarios, pulsar el botón "Save Changes" e ir a 6.</p> <p>6. Tras editar el rol se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Project Info page". Ir a 7.</p> <p>8. Tras haber editado el rol, volveremos a la pantalla de información del proyecto que contenía el rol. Podremos apreciar que los cambios realizados sobre el rol serán ya patentes.</p> <p>9. Volveremos a la pantalla de información del proyecto que contenía el rol sin que éste haya sido modificado.</p>
Excepciones	-
Frecuencia de uso	Media: esta funcionalidad se puede emplear si deseamos realizar cambios sobre los usuarios que trabajan en un rol. Podemos, a mitad del desarrollo de un proyecto, querer cambiar los usuarios que pueden desempeñar un rol. Mediante este caso de uso podemos llevarlo a cabo.

Caso de uso 3.2.3	Crear un rol EPF
Nivel de abstracción	3 Caso de uso padre: 3.2 – Modificación de roles de un proyecto
Objetivos	Creación de un nuevo rol EPF para un proyecto determinado.
Actores	Cliente vía web, sistema.
Entradas	Las características que describen el rol que son un nombre y una descripción. Además se han de asignar los usuarios del proyecto que despliegan ese rol.
Condiciones previas	Tener instanciado un proyecto.
Salidas	Confirmación de que la creación se realizó correctamente.
Post condición si éxito	La información asociada al rol EPF será introducida en la base de datos.
Post condición si fallo	-
Activador	Botón "Create Role" en la vista de listado de roles disponibles. Botón "New Role" en la vista de información de un proyecto, está situado en la parte superior junto con las demás opciones de creación.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene el rol. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus roles.</p> <p>2. En la pantalla de información del proyecto donde deseamos crear el rol, pulsamos sobre la pestaña "EPF Roles".</p> <p>3. Se mostrará el listado con todos los roles que contiene el proyecto. Elegimos el rol que queramos borrar y hacemos click sobre el botón "Create Role". Aparecerá un pequeño wizard de creación de roles</p> <p>4. Se mostrará la primera página de creación del rol. En esta introduciremos el nombre y la descripción del mismo en el campo "Brief description". También tendremos que asignarle el proyecto bajo el cual este rol tiene sentido. Cuando estemos listos, pulsar "Next", si queremos cancelar la edición pulsar "Back" e ir a 9.</p> <p>5. Tras asignar una descripción al rol, pasaremos a la pantalla de configuración de usuarios. Podemos modificar qué usuarios participan en el rol añadiéndolos con el botón "Add user" y quitándolos con el botón "Remove User". Si queremos volver a la pantalla de edición de la descripción del rol, pulsar el botón "Back" e ir a 4. Si estamos contentos con la edición de la descripción y los usuarios, pulsar el botón "Next".</p> <p>6. Tras editar el rol se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go</p>

	back to Project Info page". 7. Tras haber editado el rol, volveremos a la pantalla de información del proyecto. Podremos apreciar en la pestaña de "EpF Roles" que el nuevo rol se ha creado correctamente.
Frecuencia de uso	Media: esta funcionalidad se puede emplear si deseamos establecer nuevos roles que los usuarios pueden desempeñar y que no están contemplados por los roles clásicos (son específicos del proyecto). Podemos, a mitad del desarrollo de un proyecto, querer introducir estos nuevos roles para describir con más precisión el trabajo que desempeña un usuario. Mediante es caso de uso podemos llevarlo a cabo.

Caso de uso 3.3	Modificación de los productos de trabajo de un proyecto
Nivel de abstracción	2 Caso de uso padre: 3 – Modificación de un proyecto FVE.
Objetivos	Modificar los productos de trabajo ubicados dentro de un proyecto FVE.
Actores	Cliente vía web, sistema.
Entradas	El proyecto al que queremos realizar modificaciones sobre sus productos de trabajo.
Condiciones previas	El proyecto debe estar correctamente instanciado y tener información EPF asociada.
Salidas	Confirmación de que la modificación ha tenido éxito.
Post condición si éxito	El proyecto con sus productos de trabajo modificados.
Post condición si fallo	El proyecto no sufre modificación alguna.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Media: los productos de trabajo son el resultado de las tareas EPF sobre las que trabajan los usuarios. Por lo tanto, serán elementos que estarán en continuo cambio y revisión. Es muy probable que la modificación de los productos de trabajo se frecuente en su sistema.

Caso de uso 3.3.1	Borrar un producto de trabajo.
Nivel de abstracción	3 Caso de uso padre: 3.3 – Modificación de los productos de trabajo de un proyecto

Objetivos	Borrar un producto de trabajo del sistema.
Actores	Cliente vía web, sistema.
Entradas	El producto de trabajo que queremos borrar.
Condiciones previas	El producto de trabajo debe existir dentro de un proyecto y estar correctamente instanciado.
Salidas	Confirmación de que el borrado se realizó correctamente.
Post condición si éxito	El producto de trabajo dejará de existir en el sistema.
Post condición si fallo	No se borrará el producto de trabajo.
Activador	Icono 'papelera' junto al producto de trabajo en las pantallas en las que se lista el producto de trabajo, como por ejemplo en la pestaña "Workproducts" en la pantalla de información de proyecto.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene el producto de trabajo. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus productos de trabajo. Ir 2.</p> <p>2. En la pantalla de información del proyecto que contiene el producto de trabajo que queremos borrar, pulsamos sobre la pestaña "Workproducts". Ir a 3.</p> <p>3. Se mostrará el listado con todos los productos de trabajo que contiene el proyecto. Elegimos el producto de trabajo que queramos borrar y hacemos click sobre el icono con forma de papelera que aparece en la columna "Operations". Ir a 4.</p> <p>4. Se mostrará la pantalla de borrado del producto de trabajo con la información del mismo. Si realmente deseamos borrarlo, pulsamos el botón "Delete" y vamos a 5. Si no queremos borrarlo, pulsamos el botón "Don't delete" y vamos a 7.</p> <p>5. Tras borrar el producto de trabajo se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Project Info page". Ir a 6.</p> <p>6. Tras haber borrado el producto de trabajo, volveremos a la pantalla de información del proyecto que contenía el producto de trabajo.</p> <p>7. Volveremos a la pantalla de información del proyecto que contenía el producto de trabajo sin que éste haya sido borrado.</p>
Excepciones	-
Frecuencia de uso	Baja: el borrado de workproducts no será una operación frecuente dentro de la ejecución del proyecto.

Caso de uso 3.3.2	Editar información general de un producto de trabajo
Nivel de abstracción	3 Caso de uso padre: 3.3 – Modificación de los productos de trabajo de un proyecto
Objetivos	Editar la información general asociada a un producto de trabajo: su descripción y su nombre.
Actores	Cliente vía web, sistema.
Entradas	El producto de trabajo que queremos editar.
Condiciones previas	El producto de trabajo debe estar correctamente creado y pertenecer a un proyecto del sistema.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información general asociada al producto de trabajo quedará modificada.
Post condición si fallo	No se modificará la información general del producto de trabajo.
Activador	Enlace "Edit general information" en la pantalla de información del producto de trabajo.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del producto de trabajo. Para ello, debemos pulsar sobre el nombre del producto de trabajo donde éste aparezca en cualquier punto de la aplicación, por ejemplo en la pestaña "Workproducts" de la pantalla de información del proyecto que lo contenga. Ir 2.</p> <p>2. Una vez estemos en la pantalla de información del producto de trabajo, se listará todos los datos del workproduct en concreto. Debemos pulsar sobre el enlace "Edit general information". Ir a 3.</p> <p>3. Entraremos en la pantalla de edición de la información del producto de trabajo. Podemos modificar tanto el nombre del producto de trabajo (Name) como su descripción (Brief description). Cuando estemos satisfechos con los cambios pulsar "Save Changes" e ir a 4. Si deseamos cancelar las modificaciones realizadas, pulsar "Back" e ir a 6.</p> <p>4. Tras editar el producto de trabajo se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Workproduct Info page". Ir a 5.</p> <p>5. Volveremos a la pantalla de información del producto de trabajo donde veremos patentes los cambios realizados.</p> <p>6. Volveremos a la pantalla de información del producto de trabajo sin que éste haya sido modificado.</p>
Excepciones	-
Frecuencia de uso	Baja: tanto el nombre del producto de trabajo como su descripción son atributos del producto de trabajo que apenas sufrirán modificaciones.

Caso de uso 3.3.3	Editar información de roles y tareas de un producto de trabajo.
Nivel de abstracción	3 Caso de uso padre: 3.3 – Modificación de los productos de trabajo de un proyecto
Objetivos	Editar la información de roles y tareas asociada a un producto de trabajo.
Actores	Cliente vía web, sistema.
Entradas	El producto de trabajo que queremos editar.
Condiciones previas	El producto de trabajo debe estar correctamente creado y pertenecer a un proyecto del sistema.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información de roles y tareas asociada al producto de trabajo quedará modificada.
Post condición si fallo	No se modificará la información de roles y tareas del producto de trabajo.
Activador	Enlace "Edit roles&tasks information" en la pantalla de información del producto de trabajo.
Secuencia	<p>1. En primer lugar debemos asistir a la pantalla de información del producto de trabajo. Para ello, debemos pulsar sobre el nombre del producto de trabajo donde éste aparezca en cualquier punto de la aplicación, por ejemplo en la pestaña "Workproducts" de la pantalla de información del proyecto que lo contenga. Ir 2.</p> <p>2. Una vez estemos en la pantalla de información del producto de trabajo, se listará todos los datos del workproduct en concreto. Debemos pulsar sobre el enlace "Edit roles&tasks information". Ir a 3.</p> <p>3. Entraremos en la pantalla de edición de roles y dependencias de tareas del producto de trabajo. En la parte superior debemos elegir los roles que deberán cumplir los usuarios para trabajar en este producto de trabajo. Debemos elegir al menos un rol responsable y tantos roles adicionales como queramos. Después, en la parte inferior, configuraremos las relaciones de este producto con las tareas del proyecto al que pertenece. Sólo podemos insertar una misma tarea con el tipo de dependencia elegido una única vez, si no lo cumplimos E-1. Cuando hayamos terminado, pulsamos el botón "Next step". Si deseamos cancelar los cambios realizados, pulsar el botón "Back" e ir a 7.</p> <p>4. Ahora debemos asociar los usuarios a los roles que definimos en el anterior paso. Sólo se listarán los usuarios que cumplen el rol concreto. Cuando hayamos terminado, pulsamos el botón "Next step" y vamos a 5.</p> <p>5. Tras editar el producto de trabajo se mostrará una pantalla de</p>

	confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Workproduct Info page". Ir a 6 . 6. Volveremos a la pantalla de información del producto de trabajo donde veremos patentes los cambios realizados. 7. Volveremos a la pantalla de información del producto de trabajo sin que éste haya sido modificado.
Excepciones	E-1: se mostrará un mensaje informativo de que no se podrá insertar dos veces la misma relación tarea-workproduct con el mismo tipo de relación.
Frecuencia de uso	Media: es posible que en mitad del proceso de ejecución de un proyecto deseemos modificar los roles necesarios para implementar un producto de trabajo.

Caso de uso 3.3.4	Editar información de usuario responsable y usuarios modificadores de un producto de trabajo
Nivel de abstracción	3 Caso de uso padre: 3.3 – Modificación de los productos de trabajo de un proyecto
Objetivos	Editar la información de usuario responsable y usuarios modificadores asociada a un producto de trabajo.
Actores	Cliente vía web, sistema.
Entradas	El producto de trabajo que queremos editar.
Condiciones previas	El producto de trabajo debe estar correctamente creado y pertenecer a un proyecto del sistema.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información de usuario responsable y usuarios modificadores asociada al producto de trabajo quedará modificada.
Post condición si fallo	No se modificará la información de usuario responsable y usuarios modificadores del producto de trabajo.
Activador	Enlace "Edit users responsible and modifiers" en la pantalla de información del producto de trabajo.
Secuencia	1. En primer lugar debemos asistir a la pantalla de información del producto de trabajo. Para ello, debemos pulsar sobre el nombre del producto de trabajo donde éste aparezca en cualquier punto de la aplicación, por ejemplo en la pestaña "Workproducts" de la pantalla de información del proyecto que lo contenga. Ir 2 . 2. Una vez estemos en la pantalla de información del producto de trabajo, se listará todos los datos del workproduct en concreto. Debemos pulsar sobre el enlace "Edit users responsible and modifiers". Ir a 3 . 3. En la esta pantalla debemos asociar los usuarios a los roles que estén definidos como responsable y modificadores en el producto de trabajo. Sólo se listarán los usuarios que cumplen el rol concreto.

	<p>Cuando hayamos terminado, pulsamos el botón "Next step" y vamos a 4. Si deseamos cancelar la edición del producto de trabajo, pulsamos el botón "Back" y vamos a 6.</p> <p>4. Tras editar el producto de trabajo se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Workproduct Info page". Ir a 5.</p> <p>5. Volveremos a la pantalla de información del producto de trabajo donde veremos patentes los cambios realizados.</p> <p>6. Volveremos a la pantalla de información del producto de trabajo sin que éste haya sido modificado.</p>
Excepciones	-
Frecuencia de uso	Media: la edición de los usuarios asociados a un producto de trabajo será una acción que muy probablemente se realice con cierta frecuencia ya que eventos como la rotación de usuarios o los cambios internos pueden hacer que la distribución de los usuarios dentro del proyecto varíe.

Caso de uso 3.3.5	Crear un producto de trabajo
Nivel de abstracción	3 Caso de uso padre: 3.3 – Modificación de los productos de trabajo de un proyecto
Objetivos	Crear un nuevo producto de trabajo en un proyecto concreto.
Actores	Cliente vía web, sistema.
Entradas	Los datos del nuevo producto de trabajo a crear y el proyecto en el que deseamos crearlo.
Condiciones previas	El proyecto sobre el que vamos a crear el producto de trabajo debe estar correctamente instanciado en el sistema.
Salidas	Se muestra por pantalla un mensaje de confirmación indicando que el producto de trabajo se ha instanciado correctamente.
Post condición si éxito	Se crea dentro del proyecto elegido un producto de trabajo con los parámetros indicados por el usuario.
Post condición si fallo	No se crea el producto de trabajo.
Activador	Botón "Create workproduct" en la parte superior de la pantalla de información general de un proyecto. Botón "Create workproduct" dentro de la pestaña "Workproducts" de la pantalla de información general de un proyecto.
Secuencia	<p>1. En primer lugar debemos acudir a la pantalla de información general del proyecto al que queremos insertar el nuevo producto de trabajo. Para ello, la opción más sencilla es ir al módulo "Projects" de dotProject y pulsar sobre el nombre del proyecto elegido. Ir a 2.</p> <p>2. Una vez dentro de esta pantalla de información, debemos hacer</p>

	<p>click sobre el botón "Create workproduct" que se detalle en el apartado de activadores y pasamos a 3.</p> <p>3. La primera pantalla de creación del producto de trabajo nos permitirá insertar tanto el nombre del producto de trabajo como su descripción. También tenemos la posibilidad de elegir el proyecto en el que deseamos instanciarlo. Cuando hayamos introducido los datos, pulsamos sobre "Next step" y pasamos a 4. Si deseamos cancelar el proceso de creación del producto de trabajo, pulsamos "Back" y vamos a 8.</p> <p>4. En la siguiente página tenemos que introducir qué rol será el que deba cumplir el usuario responsable del rol así como los roles (incluido ninguno) que pueden tener los usuarios adicionales del producto de trabajo. Después, tenemos la opción de indicar qué tareas están relacionadas con este producto de trabajo y en qué tipo de relación (entrada, salida o entrada obligatoria). No podemos introducir la misma tarea dos veces con el mismo tipo de relación con el producto de trabajo, si lo hacemos ir a E-1. Cuando hayamos terminado, pulsar "Next step" y saltar a 5.</p> <p>5. En esta página tenemos que configurar que usuarios participarán en el rol basándonos en la configuración de roles que establecimos en el anterior paso. Añadiremos los usuarios los roles adicionales mediante el botón "Add user" y los eliminaremos mediante "Remove user". Cuando hayamos terminado, pulsamos el botón "Next step" y pasamos al siguiente paso, el número 6.</p> <p>6. Ya hemos terminado de crear el producto de trabajo. Se mostrará una pantalla de confirmación de que el proceso ha sido llevado con éxito. Pulsamos sobre el enlace "Go back to workproduct info page" y pasamos a 7.</p> <p>7. Veremos la pantalla de información del nuevo producto de trabajo, donde confirmaremos que la creación se ha producido con éxito.</p> <p>8. Volveremos a la pantalla de información del proyecto y no se habrá creado el producto de trabajo.</p>
Excepciones	E-1: Una ventana desplegable nos informará de que no es posible insertar la tarea dos veces con el mismo tipo de asociación y ésta no se insertará al producto de trabajo.
Frecuencia de uso	Alta: la creación de productos de trabajo es una actividad frecuente dentro de la ejecución de proyectos ya que estos funcionan como resultados palpables de las tareas.

Caso de uso 3.4	Modificación de una plantilla de un proyecto
Nivel de abstracción	2 Caso de uso padre: 3 – Modificación de un proyecto FVE.
Objetivos	Modificar las plantillas asociadas a productos de trabajo ubicados dentro de un proyecto FVE.
Actores	Cliente vía web, sistema.
Entradas	El proyecto al que queremos realizar modificaciones sobre sus

	plantillas.
Condiciones previas	El proyecto debe estar correctamente instanciado y tener información EPF asociada.
Salidas	Confirmación de que la modificación ha tenido éxito.
Post condición si éxito	El proyecto con sus plantillas modificados.
Post condición si fallo	El proyecto no sufre modificación alguna.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Media: las plantillas están diseñadas para poder recibir continuas modificaciones a medida que se van completando.

Caso de uso 3.4.1	Borrar una plantilla
Nivel de abstracción	3 Caso de uso padre: 3.4 – Modificación de una plantilla de un proyecto
Objetivos	Borrar un producto de trabajo del sistema.
Actores	Cliente vía web, sistema.
Entradas	La plantilla que queremos borrar.
Condiciones previas	La plantilla debe existir dentro de un producto de trabajo y debe estar correctamente instanciada.
Salidas	Confirmación de que el borrado se realizó correctamente.
Post condición si éxito	La plantilla dejará de existir en el sistema y su archivo asociado también será borrado.
Post condición si fallo	No se borrará la plantilla.
Activador	Icono 'papelera' junto a la plantilla en las pantallas en las que se lista la plantilla, como por ejemplo en la pestaña "Templates" de la pantalla de información de proyecto. También el mismo icono junto a la plantilla en el workproduct que la contiene.
Secuencia	1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene la plantilla. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus productos de trabajo. Ir 2. 2. En la pantalla de información del proyecto que contiene la plantilla

	<p>que queremos borrar, pulsamos sobre la pestaña "Templates". Ir a 3.</p> <p>3. Se mostrará el listado con todas las plantillas que contiene el proyecto. Elegimos la plantilla que queramos borrar y hacemos click sobre el icono con forma de papelera que aparece en la columna "Operations". Ir a 4.</p> <p>4. Se mostrará la pantalla de borrado de la plantilla con la información de la misma. Si realmente deseamos borrarla, pulsamos el botón "Delete" y vamos a 5. Si no queremos borrarlo, pulsamos el botón "Don't delete" y vamos a 7.</p> <p>5. Tras borrar la plantilla se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Project Info page". Ir a 6.</p> <p>6. Tras haber borrado la plantilla, volveremos a la pantalla de información del proyecto que contenía la plantilla.</p> <p>7. Volveremos a la pantalla de información del proyecto que contenía la plantilla sin que ésta haya sido borrada.</p>
Excepciones	-
Frecuencia de uso	Baja/Media: el borrado de plantillas no será una operación frecuente dentro de la ejecución del proyecto, aunque sí es posible que ocasionalmente se realice.

Caso de uso 3.4.2	Editar la información de una plantilla (nombre, producto de trabajo y fichero asociado)
Nivel de abstracción	3 Caso de uso padre: 3.4 – Modificación de una plantilla de un proyecto
Objetivos	Editar la información de una plantilla: su nombre, producto de trabajo al que está asociada y fichero físico asociado.
Actores	Cliente vía web, sistema.
Entradas	La plantilla que queremos editar.
Condiciones previas	La plantilla debe estar correctamente instanciada dentro de un producto de trabajo, y a su vez, dentro de un proyecto.
Salidas	Confirmación de que la edición se realizó correctamente.
Post condición si éxito	La información asociada a la plantilla quedará modificada.
Post condición si fallo	No se modificará la información de la plantilla.
Activador	Icono 'libreta' junto a la plantilla en las pantallas en las que se lista la plantilla, como por ejemplo en la pestaña "Templates" de la pantalla de información de proyecto.
Secuencia	1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene la plantilla. Para ello, hay varios métodos. El

	<p>más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus productos de trabajo. Ir a 2.</p> <p>2. En la pantalla de información del proyecto que contiene la plantilla que queremos editar, pulsamos sobre la pestaña "Templates". Ir a 3.</p> <p>3. Se mostrará el listado con todas las plantillas que contiene el proyecto. Elegimos la plantilla que queramos editar y hacemos click sobre el icono con forma de libreta que aparece en la columna "Operations". Ir a 4.</p> <p>4. Entraremos en la pantalla de edición de la libreta. Debemos elegir el producto de trabajo al que queremos que pertenezca, el nombre de la plantilla y su descripción. Igualmente, tenemos la opción de conservar el archivo que tuviera asociada la plantilla o sobrescribirlo por uno nuevo. Para ello, debemos marcar la casilla "overwrite?" y después elegir el nuevo archivo desde nuestro sistema de ficheros. Cuando estemos contentos con las ediciones, pulsamos el botón "Finish" y vamos al punto 5. Si queremos cancelar la edición de la plantilla, pulsamos el botón "Back" e ir a 6.</p> <p>5. Tras editar la plantilla se mostrará una pantalla de confirmación de que la operación se ha realizado correctamente. Pulsamos el enlace "Go back to Project Info". Ir a 6.</p> <p>6. Tras haber editado la plantilla, volveremos a la pantalla de información del proyecto que contenía la plantilla. Veremos que los cambios realizados sobre la plantilla son ya patentes.</p> <p>7. Volveremos a la pantalla de información del proyecto que contenía la plantilla sin que ésta haya sido editada.</p>
Excepciones	-
Frecuencia de uso	Media: la edición de las plantillas, sobre todo de sus archivos asociados, serán actividades que se produzcan con relativa frecuencia dentro de la ejecución del proyecto.

Caso de uso 3.4.3	Crear una plantilla
Nivel de abstracción	3 Caso de uso padre: 3.4 – Modificación de una plantilla de un proyecto
Objetivos	Crear una nueva plantilla asociada a un producto de trabajo
Actores	Cliente vía web, sistema.
Entradas	Proyecto y producto de trabajo a los que se asocia la plantilla, nombre de la plantilla, descripción y dirección del fichero físico.
Condiciones previas	Haber instanciado correctamente un proyecto que contenga al menos un producto de trabajo.
Salidas	Inserción de la plantilla en la base de datos y almacenamiento del fichero físico en el servidor.
Post condición si	Se muestra por pantalla que la operación se ha realizado con éxito .

éxito	
Post condición si fallo	Se muestra por pantalla que la operación ha sufrido algún error en su proceso de creación.
Activador	<p>Botón denominado "Create template" situado en la vista de listados de plantillas.</p> <p>Botón denominado "edit associated template info" situado en la vista de información de un producto de trabajo.</p> <p>Botón denominado "new template" situado en la vista de un proyecto. (Para este botón tenemos que residir en la vista de información de un proyecto).</p>
Secuencia	<ol style="list-style-type: none"> 1. En primer lugar debemos asistir a la pantalla de información del proyecto que contiene la plantilla. Para ello, hay varios métodos. El más sencillo es acudir al módulo "Projects" de dotProject y pulsar sobre el proyecto del que queramos conocer sus productos de trabajo. 2. En la pantalla de información del proyecto en el cual queremos crear la nueva plantilla, pulsamos sobre la pestaña "Templates". 3. En la pantalla de "Templates" buscamos el botón "Create Template" y lo pulsamos. Accedemos a un pequeño wizard de creación. 4. A continuación volvemos a seleccionar el proyecto al cual se va a asociar en la lista de desplegables y pulsamos "Next". 5. En el siguiente paso introduciremos los campos de nombre y descripción(opcional) de la plantilla y asociaremos un producto de trabajo a la plantilla. 6. Subimos el archivo de la plantilla pulsando el botón examinar. 7. Una vez completados estos pasos finalizamos el wizard pulsando de nuevo el botón "Next". 8. Visualizamos el resultado de la operación en una nueva pantalla. Excepciones E1 y E2 9. Pulsamos el botón "Go back" para retornar a la vista de listado de plantillas y observar que efectivamente se ha creado correctamente nuestra nueva plantilla.
Excepciones	<p>E1: En caso de no introducir los valores correctos, o que el nombre de la plantilla se repita la operación se abortará.</p> <p>E2: Si el proyecto tiene un repositorio asociado, el template se comitea en el mismo. Esto puede contraer problemas si la conexión entre los módulos es incorrecta y dejar colgado el explorador.</p>
Frecuencia de uso	Baja/Media: Las plantillas suelen crearse en la fase de creación de un proyecto y este caso de uso sólo contempla el caso de tratar de generar nuevas plantillas con posterioridad a esta creación.

Caso de uso 3.5	Modificación del repositorio de Subversion
Nivel de	<p>2</p> <p>Caso de uso padre: 3 – Ejecución de un proyecto FVE.</p>

abstracción	
Objetivos	Llevar a cabo operaciones de modificación sobre el estado de nuestro repositorio subversion asociado al un proyecto en cuestión.
Actores	Cliente vía web, sistema.
Entradas	Repositorio de subversion que queremos modificar.
Condiciones previas	El repositorio debe estar correctamente configurado y en funcionamiento.
Salidas	Confirmación de que la modificación del repositorio ha tenido éxito.
Post condición si éxito	El usuario logra sus objetivos de modificación cara al repositorio.
Post condición si fallo	No se consigue realizar la modificación del repositorio.
Activador	Diferentes enlaces y botones repartidos por las interfaces de la aplicación (se detallarán en los casos de usos por debajo a éste).
Frecuencia de uso	Alta: modificar el repositorio será algo que se realice muy a menudo cuando se trabaje con este sistema de control de versiones.

Caso de uso 3.5.1	Importar los archivos de un proyecto al repositorio asociado
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion.
Objetivos	Actualizar los contenidos del repositorio del proyecto instanciado.
Actores	Cliente vía web, sistema.
Entradas	Botón de acceso, dirección del fichero zip con los datos a importar.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	-
Post condición si éxito	El repositorio adquiere los contenidos del fichero comprimido
Post condición si fallo	-
Activador	Botón de "Import" situado en el panel de la derecha del menú de subversion.
Secuencia	1. Acceder al menú de subversion 2. Cargar un archivo comprimido mediante el botón "Browse"

	<p>situado a la izquierda del botón "Import"</p> <p>3. Una vez dentro del menú, pulsar el botón de "Import" situado en la parte superior del panel de la derecha.</p> <p>4. Actualización del contenido del repositorio con los archivos importados.</p> <p>5. La ventana se refrescará con los cambios realizados visualizables desde el árbol de exploración situado a la izquierda en el menú de subversion.</p>
Frecuencia de uso	Muy baja: Los usuarios necesitan acceder con poca frecuencia a esta opción para importar masivamente los contenidos de un proyecto.

Caso de uso 3.5.2	Realización de una actualización del repositorio asociado a un proyecto (Commit) Paso 1
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion
Objetivos	Actualizar los contenidos (archivos existentes y no existentes) del repositorio mediante un archivo zip con los nuevos contenidos.
Actores	Cliente vía web, sistema.
Entradas	Botón de acceso, dirección del archivo comprimido.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	
Post condición si éxito	Visualización de una página con las opciones de commit, véanse establecer un mensaje de actualización y establecer ubicación de los archivos a importar.
Post condición si fallo	-
Activador	Botón de "Commit" situado en el panel de la derecha del menú de subversion.
Secuencia	<p>1. Acceder al menú de subversion</p> <p>2. Cargar un archivo comprimido mediante el botón "Browse" situado a la izquierda del botón "Commit"</p> <p>3. Una vez dentro del menú, pulsar el botón de "Commit" situado en la parte superior del panel de la derecha.</p> <p>4. Accedemos a la ventana de commit Paso 2 descrita en la postcondición.E1</p> <p>5. Una vez descargado podemos trabajar con su contenido.</p>
Excepciones	E1: Si el archivo subido no mantiene el formato apropiado o sus contenidos son erróneos el navegador mostrará un error o se colapsará según el tipo de fallo.
Frecuencia de uso	Media/Alta: Los usuarios necesitan acceder con frecuencia a esta opción para subir los cambios que realiza en los ficheros del

	proyecto.
--	-----------

Caso de uso 3.5.3	Realización de una actualización del repositorio asociado a un proyecto (Commit) Paso 2
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion
Objetivos	Actualizar los contenidos (archivos existentes y no existentes) del repositorio mediante un archivo zip con los nuevos contenidos.
Actores	Cliente vía web, sistema.
Entradas	Botón de acceso, dirección del archivo comprimido.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Regreso a la ventana de información del proyecto con la pestaña de subversion habilitada y con el contenido actualizado de los cambios realizados en el árbol de exploración del repositorio.
Post condición si éxito	-
Post condición si fallo	-
Activador	Objetos del árbol de carpetas del repositorio asociado al proyecto.
Secuencia	<ol style="list-style-type: none"> 1. Realización de una actualización del repositorio asociado a un proyecto (Commit)Paso 1 2. Una vez dentro de la ventana de "commit", introducir el texto de información de la actualización si es necesario. En caso contrario el sistema pondrá uno por defecto. 3. Hacemos click en la carpeta del árbol de exploración del repositorio donde queremos que se actualicen los cambios 4. Regreso a la ventana de visualización del proyecto con los cambios realizados en el árbol de exploración del repositorio en la pestaña de Subversion
Frecuencia de uso	Media/Alta: Los usuarios necesitan acceder con frecuencia a esta opción para subir los cambios que realiza en los ficheros del proyecto.

Caso de uso 3.5.4	Borrar archivos o carpetas mediante el gestor de archivos de Subversion "File Manager"
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion
Objetivos	Borrar un archivo o carpeta del repositorio del sistema.

Actores	Cliente vía web, sistema.
Entradas	Vínculo con forma de icono de papelera situado a la derecha del archivo o carpeta a borrar.
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Borrado del archivo seleccionado o carpeta seleccionada en el repositorio
Post condición si éxito	La ventana se refresca mostrando el contenido actualizado del directorio que está manejando el gestor de archivos sin el archivo o carpeta.
Post condición si fallo	La ventana no muestra ningún cambio.
Activador	Icono de papelera situado a la derecha del archivo o carpeta a borrar.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Acceder al gestor de archivos "File Manager" pulsando sobre el botón "File Manager" 3. Acceder al directorio donde se desea ubicar el archivo navegando a través del árbol de exploración, o mediante los métodos de navegación del gestor de archivos "File Manager" 4. Pulsar el icono de papelera situado a la derecha del archivo o carpeta a borrar. E1.O1 5. Pulsar sí en el mensaje de confirmación emergente para borrar el archivo o carpeta. En caso contrario la operación se cancelará 6. La ventana se refresca actualizada con su nuevo contenido mostrando la nueva carpeta creada
Excepciones	E1: Si ocurre un fallo en la conexión con el módulo la operación puede quedarse atrapada en una dirección sin salida. Pulsar volver atrás en su navegador. Comunicar con el administrador el problema.
Observaciones	O1: Si se selecciona una carpeta no vacía su contenido también será suprimido del repositorio.
Frecuencia de uso	Baja/Media: Borrar contenidos de un repositorio es una tarea que se realiza con una moderada frecuencia.

Caso de uso 3.5.5	Añadir archivos al repositorio desde el gestor de archivos de subversion "File Manager"
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion
Objetivos	Subir un archivo al repositorio del sistema.
Actores	Cliente vía web, sistema.
Entradas	Botón "Upload File" y dirección local del archivo
Condiciones	Tener instanciado el proyecto correctamente en el sistema y residir

previas	en el menú de Subversion (ver caso de uso 4.2).
Salidas	Subida del archivo al sitio deseado en el repositorio del proyecto.
Post condición si éxito	La ventana se refresca mostrando el archivo subido en el directorio que está manejando el gestor de archivos.
Post condición si fallo	La ventana no muestra ningún cambio.
Activador	Botón "Upload File" situado en la parte superior de la ventana.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Acceder al gestor de archivos "File Manager" pulsando sobre el botón "File Manager" 3. Acceder al directorio donde se desea ubicar el archivo navegando a través del árbol de exploración, o mediante los métodos de navegación del gestor de archivos "File Manager" 4. Mediante el botón "Browse" (o "Examinar") elegir el archivo a actualizar o subir al repositorio. 5. Pulsar a continuación el botón "Upload File" para terminar la operación. E1 6. La ventana se refresca actualizada con su nuevo contenido mostrando el fichero subido correctamente
Excepciones	E1: Si ocurre un fallo en la conexión con el módulo la operación puede quedarse atrapada en una dirección sin salida. Pulsar volver atrás en su navegador. Comunicar con el administrador el problema.
Frecuencia de uso	Alta: Esta operación es básica en el uso de subversion, es la que permite tanto modificar como subir ficheros individualmente.

Caso de uso 3.5.6	Crear nuevas carpetas en el repositorio usando el gestor de archivos de subversion "File Manager"
Nivel de abstracción	3 Caso de uso padre: 3.5 – Modificación del repositorio de Subversion
Objetivos	Crear una nueva carpeta en el repositorio del sistema.
Actores	Cliente vía web, sistema.
Entradas	Botón "Create Folder" y nombre de la nueva carpeta a crear
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema y residir en el menú de Subversion (ver caso de uso 4.2).
Salidas	Subida del archivo al sitio deseado en el repositorio del proyecto.
Post condición si éxito	La ventana se refresca mostrando el archivo subido en el directorio que está manejando el gestor de archivos.
Post condición si	La ventana no muestra ningún cambio.

fallo	
Activador	Botón "Create Folder" situado en la parte superior de la ventana.
Secuencia	<ol style="list-style-type: none"> 1. Acceder al menú de subversion 2. Acceder al gestor de archivos "File Manager" pulsando sobre el botón "File Manager" 3. Acceder al directorio donde se desea ubicar el archivo navegando a través del árbol de exploración, o mediante los métodos de navegación del gestor de archivos "File Manager" 4. Insertar el nombre de la nueva carpeta en el campo de texto asociado al botón "Create Folder". 5. Pulsar a continuación el botón "Create Folder" para terminar la operación. E1 6. La ventana se refresca actualizada con su nuevo contenido mostrando la nueva carpeta creada
Excepciones	E1: Si ocurre un fallo en la conexión con el módulo la operación puede quedarse atrapada en una dirección sin salida. Pulsar volver atrás en su navegador. Comunicar con el administrador el problema.
Frecuencia de uso	Alta: Crear carpetas es una tarea esencial para tener bien distribuido el contenido del repositorio.

3.3.4. Modificación de los parámetros del módulo de Subversion

Caso de uso 4	Modificación de los parámetros del módulo de Subversion
Nivel de abstracción	1
Objetivos	Modificar la configuración actual de Subversion en el sistema.
Actores	Cliente vía web, sistema.
Entradas	Configuración actual del módulo de subversion.
Condiciones previas	Necesitamos conocer la nueva configuración a aplicar sobre subversion para que éste pueda funcionar correctamente con la nueva configuración.
Salidas	Confirmación de los cambios realizados sobre la configuración.
Post condición si éxito	La configuración que permite a nuestro módulo de subversion funcionar correctamente está ajustada en base a las modificaciones introducidas por el usuario.
Post condición si fallo	No hay alteraciones sobre la configuración del módulo de subversion.
Activador	Distintos activadores que se detallan en los casos de uso que derivan de éste.

Frecuencia de uso	Muy baja. La configuración de subversion será sólo modificada sólo al principio, tras crear un nuevo repositorio SVN al que queremos enlazar o en casos muy concretos en los que la configuración actual del repositorio varíe en mitad de la ejecución de un proyecto.
-------------------	---

Caso de uso 4.1	Acceder al panel de configuración de los componentes de Subversion
Nivel de abstracción	2 Caso de uso padre: 4 – Modificación de los parámetros del módulo de Subversion
Objetivos	Configurar las características propias de subversion y realizar su activación.
Actores	Cliente vía web, sistema.
Entradas	-
Condiciones previas	-
Salidas	<p>Página compuesta por un panel de control con las siguientes opciones:</p> <ul style="list-style-type: none"> • Muestra el estado de subversion (activo o no) y a través de un botón permite realizar la activación o desactivación de subversion • Muestra el puerto de internet mediante el cual podemos acceder al módulo JSP de subversion y su correspondiente botón de cambiar la propiedad. • Muestra el puerto de internet mediante el cual podemos acceder al módulo JSP de subversion y su correspondiente botón para cambiar la propiedad. • En caso de que nunca hayamos creado un repositorio local mediante esta interfaz. <p>- Si subversion está configurado como activo, entonces se muestra una pantalla de creación de un repositorio nuevo.</p>
Post condición si éxito	Se muestra por pantalla la página de configuración de subversion descrita en Salidas.
Post condición si fallo	-
Activador	Para acceder al panel de configuración de subversion tenemos dos maneras: acceder mediante el menú superior de FVEProjectManager en el enlace marcado como subversion; o bien, si al tratar de acceder a la pantalla de visualización de subversion (caso de uso 2.5.1), subversion reside inactivo, entonces se muestra un vínculo destino esta página.
Secuencia	1. Mediante los activadores descritos en el anterior punto, accedemos a la página de configuración de subversion.

Frecuencia de uso	Muy baja: Realizar la configuración de subversion se suele hacer solamente la primera vez que se instala el FVEProjectManager en el sistema
-------------------	---

Caso de uso 4.2	Modificar el estado de subversion
Nivel de abstracción	2 Caso de uso padre: 4 – Modificación de los parámetros del módulo de Subversion
Objetivos	Establecer la disponibilidad del sistema del módulo anexo de JSP que permite realizar las operaciones sobre subversion
Actores	Cliente vía web, sistema.
Entradas	Botón de activación/desactivación
Condiciones previas	Tener instanciado el proyecto correctamente en el sistema.
Salidas	Modificación de la propiedad en la ventana del panel de configuración de subversion.
Post condición si éxito	Cambio del estado de disponibilidad del módulo de subversion a activado o desactivado según proceda.
Post condición si fallo	-
Activador	Botón "Activate" / "Deactivate" situado en la página de configuración de subversion.
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de configuración de subversion haciendo click en el enlace situado en el menú superior del módulo FVEProjectManager. 2. Una vez dentro de la página configuración de subversion, aparecerá un botón de "actívale" o "deactívale" el módulo. 3. Se aplicarán los cambios en la base de datos para impedir el acceso a este módulo desde cualquier punto de FVEProjectManager. 4. Se mostrará el estado actualizado del módulo en la ventana de configuración de subversion.
Excepciones	-
Frecuencia de uso	Baja: se suele activar subversion una sola vez, en el momento de la instalación del FVEProjectManager .

Caso de uso 4.2.1	Modificar la cadena de enrutamiento de subversion
Nivel de abstracción	3 Caso de uso padre: 4.2 – Modificar el estado de subversion

Objetivos	Establecer el puerto de enrutamiento de las páginas que tienen vínculos al módulo JSP.
Actores	Cliente vía web, sistema.
Entradas	Campo de texto con el valor del puerto de enrutamiento y botón de envío de la propiedad adyacente al campo de texto con nombre "Apply".
Condiciones previas	-
Salidas	Modificación de la propiedad en la ventana del panel de configuración de subversion.
Post condición si éxito	Cambio del estado de puerto de enrutamiento al módulo de j2ee de subversion.
Post condición si fallo	-
Activador	Botón "Apply" situado en la página de configuración de subversion a la derecha del texto "Subversion Port".
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de configuración de subversion haciendo click en el enlace situado en el menú superior del módulo FVEProjectManager. 2. Una vez dentro de la página configuración de subversion, aparecerá un campo de texto a la derecha del texto "Subversion related url". Rellenarlo con el valor deseado. 3. Pulsar el botón "Apply" a la derecha del campo de texto modificado. 4. Se aplicarán los cambios en la base de datos para permitir el acceso a este módulo desde cualquier punto de FVEProjectManager. 5. Se mostrará el estado actualizado del módulo en la ventana de configuración de subversion.
Excepciones	-
Frecuencia de uso	Baja: se suele activar subversion una sola vez, en el momento de la instalación del FVEProjectManager .

Caso de uso 4.2.2	Modificar el puerto de enrutamiento de subversion
Nivel de abstracción	3 Caso de uso padre: 4.2 – Modificar el estado de subversion
Objetivos	Establecer el puerto de enrutamiento de las páginas que tienen vínculos al módulo JSP.
Actores	Cliente vía web, sistema.
Entradas	Campo de texto con el valor del puerto de enrutamiento y botón de envío de la propiedad adyacente al campo de texto con nombre "Apply".

Condiciones previas	-
Salidas	Modificación de la propiedad en la ventana del panel de configuración de subversion.
Post condición si éxito	Cambio del estado de puerto de enrutamiento al módulo de j2ee de subversion.
Post condición si fallo	-
Activador	Botón "Apply" situado en la página de configuración de subversion a la derecha del texto "Subversion Port".
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de configuración de subversion haciendo click en el enlace situado en el menú superior del módulo FVEProjectManager. 2. Una vez dentro de la página configuración de subversion, aparecerá un campo de texto a la derecha del texto "Subversion Port". Rellenarlo con el valor deseado. 3. Pulsar el botón "Apply" a la derecha del campo de texto modificado. 4. Se aplicarán los cambios en la base de datos para permitir el acceso a este módulo desde cualquier punto de FVEProjectManager. 5. Se mostrará el estado actualizado del módulo en la ventana de configuración de subversion.
Excepciones	-
Frecuencia de uso	Baja: se suele activar subversion una sola vez, en el momento de la instalación del FVEProjectManager .

Caso de uso 4.3	Creación del repositorio local
Nivel de abstracción	2 Caso de uso padre: 4 – Modificación de los parámetros del módulo de Subversion
Objetivos	Crear un repositorio de subversion en el sistema para permitir que los contenidos de los proyectos puedan gestionarse mediante esta aplicación en el sistema.
Actores	Cliente vía web, sistema.
Entradas	-
Condiciones previas	<ol style="list-style-type: none"> 1. Haber configurado correctamente los valores de enrutamiento (puerto, y dirección relativa) al módulo j2ee de subversion. 2. Haber activado el módulo en el menú de configuración
Salidas	Creación de un repositorio en el sistema físico local.
Post condición si	Esta opción desaparece del menú de configuración del módulo.

éxito	
Post condición si fallo	Esta opción sigue permanente en el sistema, el repositorio se podrá intentar crear más adelante.
Activador	Botón "Create Subversion Repository" situado en la parte inferior del segundo encuadre de la página de configuración de subversion.
Secuencia	<ol style="list-style-type: none"> 1. Acceder a la página de configuración de subversion haciendo click en el enlace situado en el menú superior del módulo FVEProjectManager. 2. Pulsar el botón "Create Subversion Repository" situado en la parte inferior de la página. 3. Se crea el repositorio en el sistema local permitiendo su uso desde la aplicación. Si fracasa ir a E1 4. La opción es deshabilitada del menú de configuración de subversion.
Excepciones	E1: En caso de que la configuración del módulo j2EE de FVE no esté correctamente instalada, o la configuración de acceso a este mediante FVE el repositorio no se instalará en el sistema. El sistema volverá al sistema anterior si fue un fallo del módulo ó bien el explorador se situará en una página errónea.
Frecuencia de uso	Baja: se suele activar subversion una sola vez, en el momento de la creación del repositorio local.

4. Desarrollo

4.1. Dónde termina dotProject y empieza FVE

Todo el código de nuestra aplicación se asienta sobre el código de dotProject. Es por ello importante realizar una clara distinción entre las partes de dotProject y las partes de FVE Project Manager.

4.1.1. Qué funcionalidades ofrece dotProject

DotProject es un gestor de proyectos autónomo que permite la creación de proyectos, ejecución y terminación de los mismos. Ofrece soporte para diferentes elementos de los proyectos con los que trabaja. Para cada uno de ellos, utiliza un módulo autónomo que contiene todos los archivos necesarios para hacer funcionar estos elementos. A continuación se perfilarán los principales módulos que utiliza dotProject [dPDocumentation108], entendiéndose así que funcionalidades ofrece dotProject a sus usuarios:

- **Projects:** es uno de los módulos principales de dotProject. Realiza la gestión de los proyectos del sistema. Permite la creación, borrado, visualización y edición de las características de los proyectos. También tiene archivos para la generación de informes relacionados con los proyectos instanciados en el sistema.
- **Tasks:** las tareas tienen casi tanta importancia como los proyectos. Las tareas tienen un fin determinado que los usuarios asociados a ellas deben cumplimentar. Este módulo se encarga de hacer funcionar correctamente las tareas y permitir realizar todas las operaciones de gestión comunes sobre ellas como visualización, creación, edición o borrado.
- **Companies:** este módulo contiene todos los archivos precisos para la gestión de compañías. Cada compañía puede tener asociados proyectos y usuarios. De esta forma, es mucho más fácil clasificar los proyectos y usuarios por la empresa para los que estos trabajan.
- **Admin:** módulo de administración de dotProject. Permite el ajuste de todas las variables de configuración que emplea dotProject.
- **Calendar, forum, file:** distintos módulos que realizan diversas funciones auxiliares que facilitan la ejecución de los proyectos, como calendarios para resaltar las fechas, foros de intercomunicación entre usuarios y subida y bajada de archivos simples al servidor del programa.

Al implementar nuestra aplicación, contamos con la ventaja de que toda esta funcionalidad ya estaba implementada con un funcionamiento probado y muy satisfactorio. Evitábamos de esta forma tener que implementar

funciones necesarias en la gestión de proyectos como la edición, creación y borrado de tareas y proyectos.

Optamos por añadir todo el grueso de nuestra aplicación como un módulo más de dotProject, es decir, de igual forma que trabajan los módulos citados anteriormente. Sin embargo, la inclusión de la funcionalidad extra por parte de nuestra aplicación requería de la modificación en algunos elementos de dotProject para enlazar de forma directa nuestras nuevas características con las que ya ofrecía dotProject. Se detallará este aspecto más en detalle en la sección 4.1.3.

4.1.2. Diferencias entre elementos de dotProject y de FVE

En FVE Project Manager se incluyen algunos elementos que pueden causar confusión con los ya existentes en dotProject si no se realiza una matización clara. Seguidamente se mostrarán las diferencias entre estos elementos utilizados en la gestión de proyectos de FVE Project Manager.

- **Proyectos y proyectos FVE:** los proyectos sin indicativo hacen referencia a los proyectos normales que utiliza dotProject. Nuestro módulo es totalmente compatible con toda la estructura y elementos de dotProject, por lo que podremos seguir trabajando con proyectos normales de la misma forma en la que se hace con dotProject. Los proyectos FVE sin embargo son los proyectos que utiliza nuestro sistema. Tienen una base idéntica a los proyectos empleados por dotProject, salvo que se le añada nueva funcionalidad tal como información asociada al archivo XML importado o los usuarios que trabajan en el proyecto. Estos usuarios se asocian de forma diferente a la que se utiliza en los proyectos dotProject normales.
- **Tareas y tareas EPF:** las tareas 'normales' son las que utiliza y gestiona dotProject. De nuevo y al igual que con los proyectos, nuestro módulo es compatible con el funcionamiento de estas tareas. De hecho, la mayor parte de la funcionalidad ofrecida por dotProject es reutilizada en las tareas EPF, como por ejemplo la creación, modificación o borrado de las mismas. Estas tareas EPF son análogas a las tareas normales de dotProject, salvo que añaden nueva información EPF: tipo de tarea, usuarios que trabajan en ella y con qué roles EPF asignados. Las tareas EPF tienen un rol responsable y varios (incluido ninguno) roles adicionales. Para estos roles, debemos asociar usuarios que los desempeñen: un usuario para el rol responsable y cero o más usuarios para cada rol adicional. Esta asociación es totalmente independiente a la estructura de dotProject.
- **Roles y roles EPF:** la gestión de roles que ofrece dotProject por defecto es muy pobre y no ofrece demasiada funcionalidad aparte del rol 'admin' que tiene derecho de administración en el sistema.

Nuestro módulo FVE integra roles EPF, que son unos roles independientes a los que utiliza dotProject. Tienen asociado un nombre y una descripción, al igual que los roles estándar. Sin embargo, estos nuevos roles tienen como fin el ser asignados como responsable o adicionales de las tareas o primario o modificador de los productos de trabajo. Además, son importados directamente desde la definición de proceso que da forma a la instancia de nuestro proyecto.

Los usuarios que se emplean en FVE son los mismos que utiliza dotProject. Los productos de trabajo, las plantillas y las características de Subversion son elementos añadidos sobre dotProject que no generan ninguna incompatibilidad.

4.1.3. Dónde se modifica el código base de dotProject

Dentro de nuestro módulo se ofrece toda la funcionalidad necesaria para que todos los requisitos sean capturados. Sin embargo, era necesario lograr una integración más directa con dotProject, de forma que sobre la estructura base de dotProject pudieran intercalarse los nuevos elementos que aporta nuestro módulo FVE. Por esta razón, llevamos a cabo la modificación de archivos base de los módulos de dotProject. A continuación, se muestra una lista con los ficheros de dotProject que fueron modificados y la razón para llevar a cabo estas modificaciones.

- **/modules/calendar/day_view.php:** sirve para mostrar un informe diario de todas las actividades que debe llevar a cabo el usuario. Es la primera página que se carga al hacer login al sistema. Por ello, realizamos la modificación pertinente para que al cargar este archivo se llevara a cabo la inicialización de la sesión FVE (ver 5.4.2.2) de la que se hará uso durante el resto de la sesión del usuario en el sistema.
- **/modules/admin/viewuser.php:** este fichero se utiliza para visualizar la pantalla de información del usuario. Sin embargo, dotProject no muestra por defecto en qué productos de trabajo trabaja el usuario o qué roles EPF puede desempeñar o está actualmente desempeñando. Para añadir esta información de EPF, realizamos la modificación del archivo para incluir 4 nuevas pestañas que informaran del estado respecto a los elementos EPF del usuario.
- **/modules/projects/index.php:** lista todos los proyectos existentes en el sistema. Se lleva a cabo la adición de un texto para indicar que el módulo FVE está activo. También se incluye un botón para la creación de proyectos FVE a partir del wizard de creación de proyectos. También se añade la pestaña 'FVE Projects' para poder filtrar sólo los proyectos con funcionalidad EPF asociada.

- **/modules/projects/reports.php**: este archivo lista todos los posibles informes que se pueden aplicar a un proyecto. Se añade el nuevo tipo de informe que incluimos en la aplicación: "Project Statistics: Include FVE Stats".
- **/modules/projects/reports/stats.php**: nuevo informe añadido en el proyecto que informa sobre el estado del proyecto teniendo en cuenta todos los elementos EPF que introduce nuestro módulo.
- **/modules/projects/view.php**: genera la pantalla de visualización de un proyecto. Se añade un recuadro que contiene toda la nueva información EPF de nuestro proyecto y cuatro nuevas pestañas con información de productos de trabajo, plantillas, roles y subversion relacionada con el proyecto actual.
- **/modules/projects/vw_idx_fveprojects.php (Añadido)**: nueva pestaña que se muestra en la página que lista todos los proyectos del sistema. Mostrará únicamente los proyectos con funcionalidad EPF asociada.
- **/modules/tasks/index.php**: archivo principal que lista todas las tareas del sistema. Se añade en la parte superior un enlace que indica si el módulo FVE está correctamente funcionando.
- **/modules/tasks/view.php**: página de visualización de la información de la tarea. Se añade un recuadro con la información EPF de la tarea. En caso de que la tarea no tenga información EPF asociada, no se producirá ningún conflicto.
- **/style/main.css**: sobre el archivo CSS principal que utiliza dotProject para dar formato de presentación a sus páginas web, se añaden tres nuevos estilos para la presentación de algunas tablas que utilizamos en nuestro módulo.

Con estas modificaciones, la integración entre nuestro módulo y dotProject es suficientemente estrecha como para permitir una interacción cómoda del usuario con las funcionalidades EPF que ofrece nuestra aplicación.

4.2. Limitaciones y convenios a seguir de dotProject

La base de datos utilizada para las pruebas ha sido MySQL, aunque se han realizado consultas genéricas utilizando clases propias de dotProject para permitir el uso de otros tipos de bases de datos basados en SQL soportados por dotProject y favorecer la flexibilidad del sistema. Por ello y por la similitud de los entorno SQL, la arquitectura es independiente del sistema gestor de base de datos empleado posteriormente. Estos convenios adoptados se pueden resumir en:

- Uso de la clase interna DBQuery: Esta clase encapsula las consultas a la base de datos, proporcionando una estructura completa para realizar consultas mediante objetos. El uso de esta clase se decide por la mayor claridad del código generado, facilitando las labores de depuración, por uniformidad con el resto de módulos de dotProject y para generar código que sea fácilmente entendible por otros desarrolladores de dotProject. Debido a ser algo diferente a las sentencias SQL se realizó un pequeño tutorial sobre su uso para adaptarse a sus estructura.
- Carga de scripts mediante el sistema interno de dotProject `InstallLoadSql(script.sql)`. Esto posibilita el informe de errores automático de dotProject al crear las bases de datos o realizar alguna otra función que requiera del uso de un script.
- Uso de la sesión global cuando sea posible: Esta práctica intenta por una parte no complicar la información que debe manejar el módulo, y por otro favorecer la flexibilidad del sistema. El sistema, por ejemplo, almacena en la sesión el usuario, contraseña y nombre de base de datos y garantizar el funcionamiento en cualquier entorno implica utilizar estos datos para realizar conexiones implícitas a la base de datos. Para llamarse a esta información se utilizarán primitivas de la forma `dPgetParam('nombre_parámetro')`.
- En general, aunque no todos los aspectos tuvieron influencia en nuestra aplicación, se intentaron respetar los convenios de programación que figuran en el documento Coding Standard de la wiki de dotProject [`dPDocumentation2 07`].

4.3 Arquitectura del sistema

En este esquema se explicará la estructura interna de nuestra aplicación. Se desglosará en diferentes puntos para tratar de dar una visión conceptual que se corresponda con el funcionamiento interno del programa.

4.3.1 Diagramas de despliegue

Un diagrama de despliegue es un esquema que se emplea para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Nuestra aplicación, al utilizar diferentes servicios que pueden ejecutarse en máquinas diferentes, puede dar lugar a múltiples despliegues posibles, aunque nosotros sólo analizaremos las que hemos considerado más importantes. A continuación explicaremos los esquemas principales que se pueden desplegar para utilizar nuestra aplicación en entornos de área local o a través de internet.

4.3.1.1 Despliegue en LAN (Local Area Network) con Base de Datos Integrada

Este despliegue es uno de los más típicos. Los clientes se conectarán al servidor de dotProject + FVE que se encuentra dentro de nuestra red de área local. El acceso se realiza sin problemas introduciendo la dirección IP del equipo que tiene los procesos corriendo. Si se posee un servidor de DNS, podemos introducir directamente el nombre del servidor en el navegador web para acceder. Este servidor alberga todos los servicios en la misma máquina. En la siguiente imagen se sintetiza este esquema:

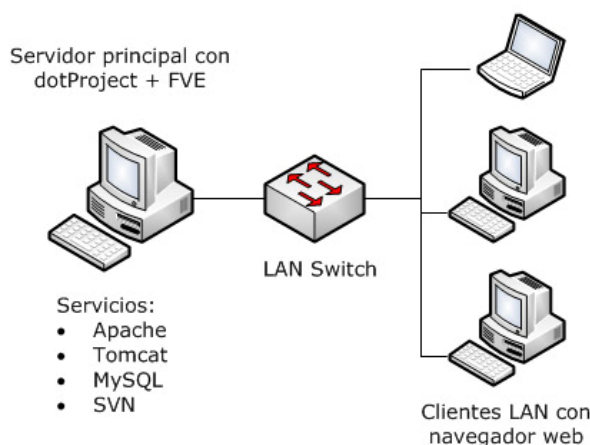


Diagrama de despliegue. Esquema A.

4.3.1.2 Despliegue en LAN (Local Area Network) con Base de Datos Separada

Este esquema es muy similar al anterior, pero en muchos entornos empresariales es común tener servidores dedicados para las bases de datos. DotProject permite sin problemas direccionar la base de datos a un servidor externo al que se utiliza para visualizar y compilar las páginas web. De cara al usuario, esta modificación es transparente y utilizando este despliegue se pueden ganar muchas ventajas como optimización de la velocidad o mejora de la seguridad. A continuación se muestra un esquema de esta organización para entender mejor el concepto:

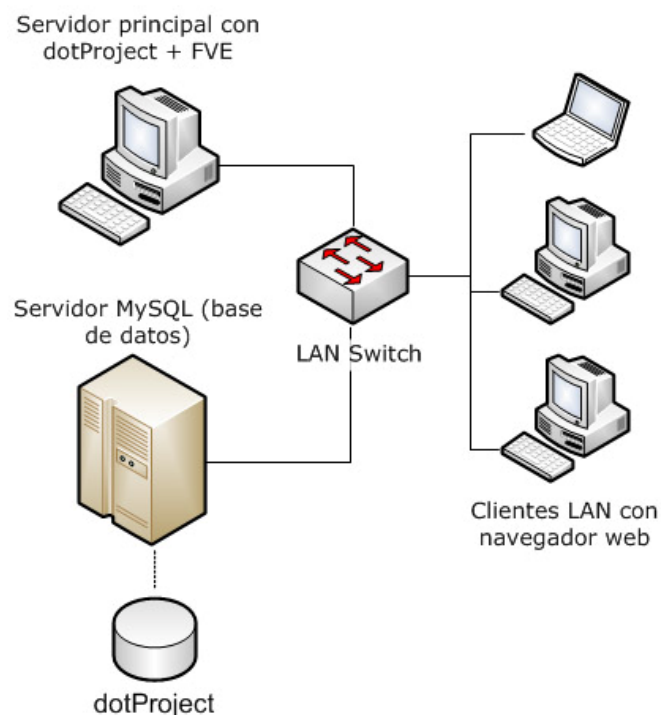


Diagrama de despliegue. Esquema B.

4.3.1.3 Despliegue en WAN (Wide Area Network) con Base de Datos Separada

La tercera alternativa es también muy común por su flexibilidad y potencia. Al funcionar bajo un servidor web, dotProject + FVE puede ser perfectamente accedido a través de Internet. Esto ofrece las ventajas de que los usuarios puedan conectarse desde cualquier punto e incluso desde la misma red local trabajando siempre sobre la misma instancia de la aplicación. Los únicos requisitos necesarios para esta implementación es obtener una dirección IP pública y fija (o dinámica, pero no es recomendable) en la conexión que tenga el servidor que albergue dotProject de manera que desde cualquier ubicación, con introducir esta dirección IP en nuestro navegador, nos dirija a la página de nuestra aplicación. En el siguiente diagrama se puede apreciar un ejemplo de este tipo de despliegue:

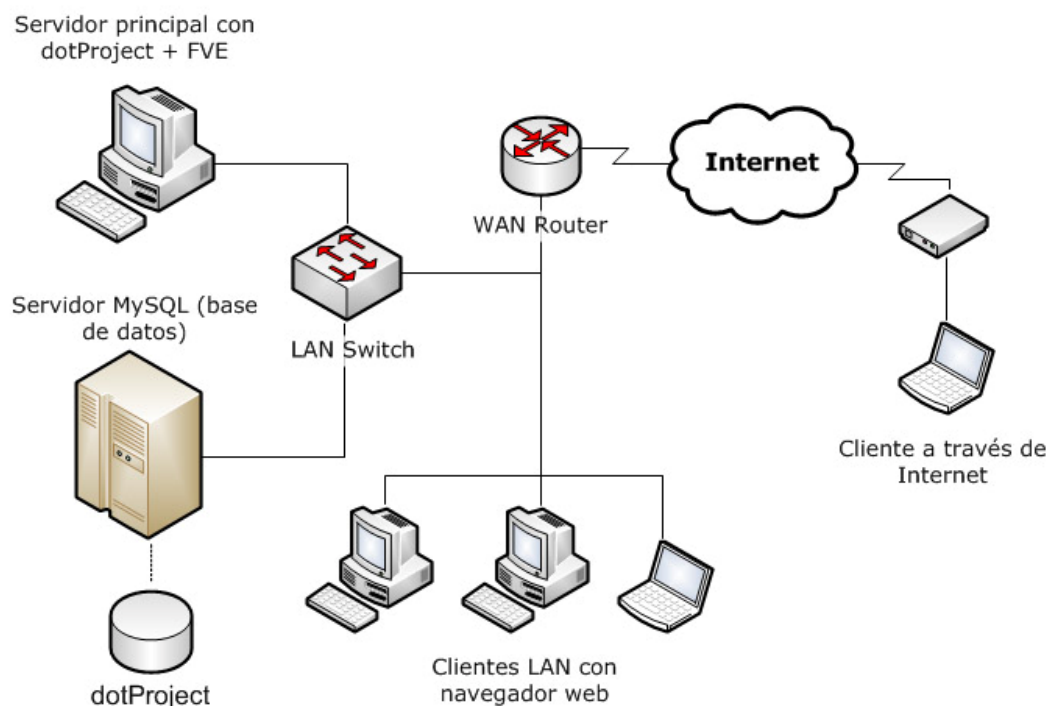


Diagrama de despliegue. Esquema C.

4.3.2 Diseño del sistema

Es preciso realizar una pequeña matización antes de describir el diseño de nuestro sistema. La aplicación está escrita principalmente en 4 lenguajes: PHP, HTML, Java Y JavaScript. Java se utiliza en la parte concerniente a Subversion, y se compila bajo el servidor de aplicaciones Tomcat. Los otros tres lenguajes restantes están orientados a la producción de páginas web, que serán la vía de comunicación con el usuario. Es decir, las interfaces de nuestra aplicación serán páginas web con las que el usuario recibirá la información y podrá interactuar para enviar datos a la aplicación. Las aplicaciones web tienen un diseño que varía respecto de las aplicaciones stand-alone tradicionales, como pueden ser las desarrolladas en Java o .Net. Esto se debe a las restricciones que impone este tipo de desarrollo, ya que es necesario generar distintas páginas web de cara a la visualización por parte del usuario. Además, al implementar nuestra aplicación a partir de dotProject, tuvimos que adaptarnos a la forma en la que éste estaba creado.

PHP ofrece funcionalidades de Programación Orientada a Objetos y, aunque no hemos centralizado todo nuestro desarrollo en este paradigma, hemos hecho uso de estas herramientas facilitadas por PHP para implementar algunos aspectos de nuestra aplicación donde creímos que su utilización serviría para mejorar la estructura general del programa.

En los siguientes puntos se hablará de la estructura de archivos utilizada en nuestro programa así como de la utilización de clases dentro de nuestro diseño.

4.3.2.1 Mapa y funcionamiento estructural de la aplicación

En este apartado vamos a describir la estructura que sigue nuestro sistema y cómo se llevan a cabo entre sí las llamadas entre las diferentes páginas. Lo principal es conocer cuál es la base de funcionamiento del programa.

Arquitectura basada en módulos

El siguiente gráfico muestra mediante un diagrama de componentes UML la estructura básica de dotProject (sin estar incluidos todos los módulos) a la izquierda y los componentes que intervienen en FVE a la derecha. Como se puede apreciar, se respeta el núcleo principal de dotProject y únicamente se realizan modificaciones sobre los módulos de tareas, proyectos y se introduce un nuevo módulo, FVE, que es el que contiene todos los archivos para gestionar las nuevas funcionalidades que utiliza nuestra aplicación.

Diagrama componentes dotProject Original

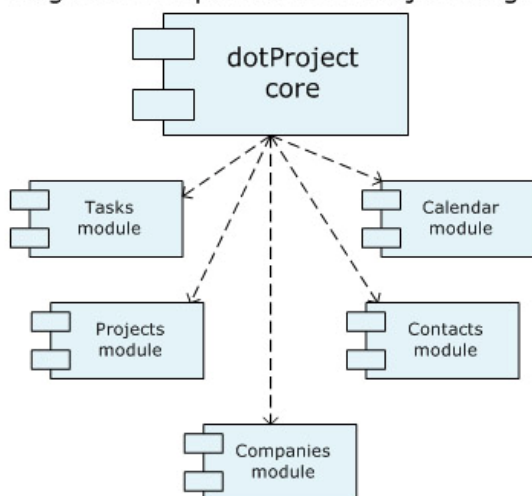
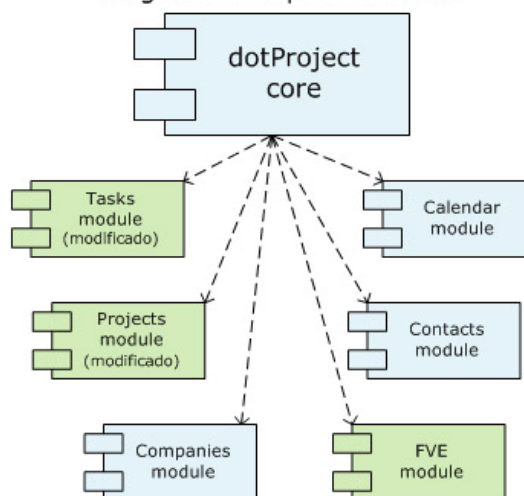


Diagrama componentes FVE



Comparación entre la estructura de dotProject y la de FVE Project Manager

DotProject utiliza módulos para repartir su funcionalidad en unidades independientes más pequeñas y manejables. Nuestra aplicación introduce un nuevo módulo a los ya utilizados por dotProject. En este módulo se incorporan todos los archivos necesarios que necesita nuestra aplicación para funcionar. Pese a la inclusión de este módulo sobre dotProject, debido al carácter de nuestra aplicación, nos era necesario introducir una serie de cambios sobre la propia estructura de dotProject (más en concreto sobre los módulos que trae de forma predefinida) para poder integrar la nueva

funcionalidad que aporta nuestro módulo en las interfaces que dotProject utiliza por defecto. Este módulo se desglosará y explicará más en detalle en el siguiente apartado.

Pese a realizar esta invasión sobre el código de dotProject, bien es cierto que estas modificaciones se han limitado lo máximo posible y su fin es únicamente el de mejorar la interacción del usuario con la funcionalidad añadida por nuestro módulo sobre dotProject.

Estructura de archivos de la aplicación

Cuando instalamos FVE Project Manager, el sistema ya está configurado para empezar a trabajar con él. En este punto vamos a explicar cuál es la estructura base dentro de nuestro módulo y cómo funciona el direccionamiento de páginas.

La siguiente imagen muestra el diagrama de componentes bajo nuestro módulo FVE, que es el que controla toda la funcionalidad añadida sobre dotProject:

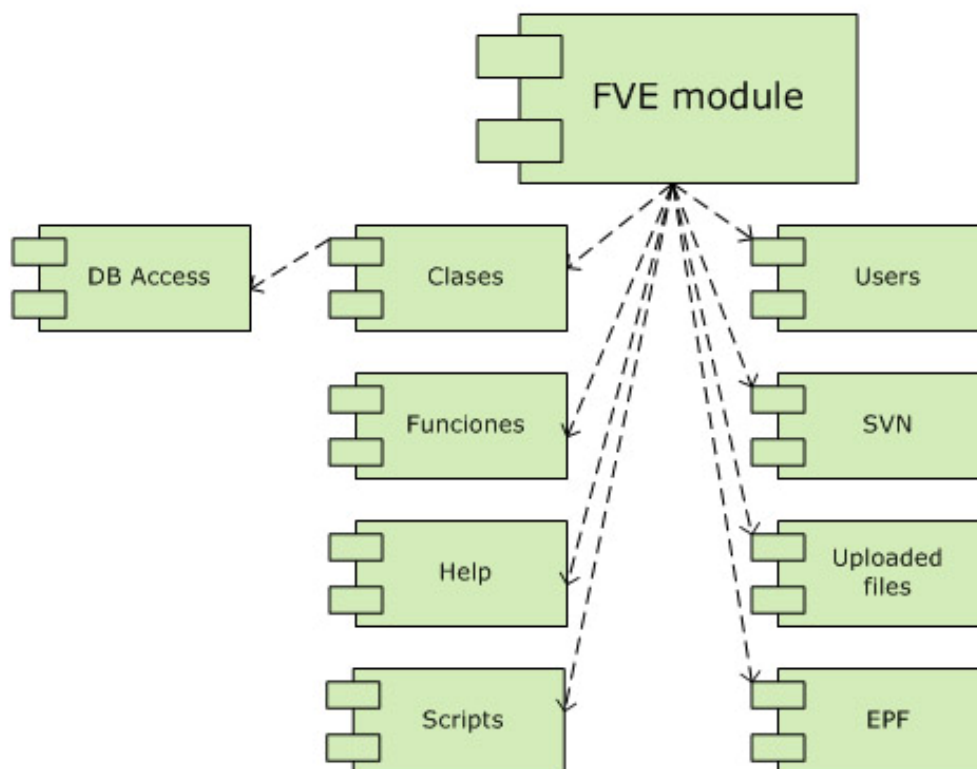


Diagrama de componentes del módulo FVE utilizado por nuestra aplicación

Para cada componente se ha creado una carpeta para facilitar así su modularidad.

Existen 7 carpetas, cada una con un uso específico que explicaremos a continuación:

- **/Clases:** esta carpeta, como su nombre indica, contiene todos los archivos de clases que utiliza FVE. El paradigma empleado a la hora de desarrollar el módulo no es el de programación orientada a objetos, pero hemos hecho uso de clases en algunos apartados de la aplicación para facilitar algunos aspectos, como el acceso a las tablas de las bases de datos, o para mantener información de la sesión durante el tiempo que el usuario esté conectado. Se detallarán en profundidad en el apartado 4.3.2.2.
- **/Funciones:** muchas veces se requiere realizar funciones auxiliares dentro del código de la aplicación. Para evitar redefinir todas estas funciones en las páginas donde se utilizan, creamos la carpeta funciones que contiene archivos PHP que contienen funciones auxiliares. Dentro de esta carpeta, existen nuevas subcarpetas para categorizar los archivos de funciones en función de su utilidad.
- **/Help:** esta carpeta contendrá el manual de usuario en formato HTML. El fin de realizar el manual en este formato es poder acceder a él a través de la misma aplicación, únicamente abriendo una nueva ventana donde aparecerán sus contenidos. La versión adaptada de este manual podrá encontrarse en el apéndice C del presente documento.
- **/Scripts:** contiene los scripts de creación y destrucción de las tablas extra sobre dotProject que necesita FVE para poder funcionar. Este script es llamado desde la página de configuración que se ejecuta cuando se instala el módulo. Igualmente, cuando se desinstala el módulo FVE, se llama al script correspondiente para dejar el estado de la base de datos de forma coherente.
- **/Users:** alberga los archivos necesarios para la gestión de usuarios dentro de FVE. Pese a que el soporte prestado a los usuarios es limitado (únicamente mostrar un listado con las tareas y proyectos EPF activos), era necesario estructurarlo correctamente dentro de nuestro módulo. No está incluido dentro de la carpeta EPF ya que los usuarios son elementos que utiliza ya por defecto dotProject y no son elementos nuevos que introduce EPF.
- **/SVN:** contiene todos los archivos necesarios para hacer funcionar todos los componentes de conexión con el servidor Subversion. Aquí se encontrarán tanto los ficheros PHP que generan las interfaces de FVE que conectan con subversion, como el repositorio y los archivos temporales que se gestionan por parte de Subversion. (Obs: No ha de estar incluido forzosamente el código de la parte de j2EE en este directorio debido a que normalmente se establece en el directorio de despliegue de un servidor como Tomcat, etc.).

- **/Uploaded Files:** en realidad se trata de dos carpetas diferentes. En una de ellas se almacenan temporalmente los archivos XML que el usuario puede subir para definir sus nuevos proyectos. La otra se utiliza para almacenar los archivos de plantilla que el usuario sube al servidor para asociar a un producto de trabajo.
- **/EPF:** esta carpeta contiene todos los recursos necesarios para hacer funcionar la nueva funcionalidad EPF que introduce FVE Project Manager sobre dotProject. Debido a su estructura, es mejor hacer un desglose y tratarla de manera independiente.

La carpeta EPF contiene en su interior nuevos archivos estructurados de nuevo con un esquema de carpetas como el que aparece en la siguiente ilustración:

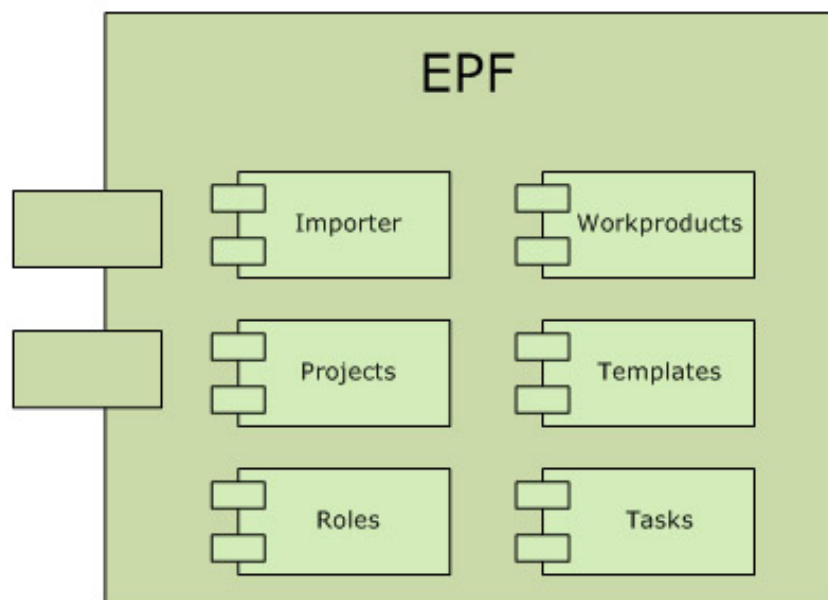


Diagrama de componentes del componente EPF ubicado dentro del módulo FVE

Seguidamente vamos a analizar de forma independiente cada una de las carpetas que contiene EPF:

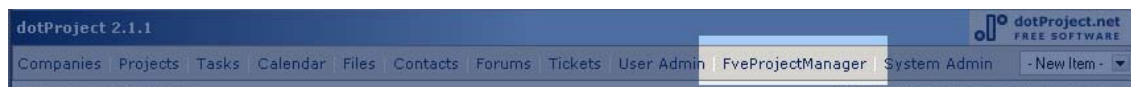
- **/EPF/Importer:** contiene los archivos PHP que se encargan de realizar la importación de los ficheros XML creados con EPF Composer suministrados por el usuario. Estos archivos XML son parseados para obtener de ellos la información necesaria para conformar la estructura de tareas, productos de trabajos, plantillas y roles de los nuevos proyectos a instanciar.
- **/EPF/Projects:** alberga archivos relacionados con la gestión de proyectos FVE. También contiene los ficheros que gestionan el Wizard de creación de proyectos y todos los pasos que éste utiliza para instanciar un nuevo proyecto a partir de un archivo XML.

- **/EPF/Roles:** aquí están ubicados todos los archivos que controlan los roles EPF que se utilizan en FVE Project Manager. Contienen archivos para la generación de informes, operaciones de borrado, creación y modificación.
- **/EPF/Tasks:** contiene archivos PHP para la gestión de las operaciones relacionadas con la información adicional que añade FVE Project Manager a las tareas. Gracias a los archivos contenidos en esta carpeta se permiten las operaciones de modificación de la información EPF de las tareas.
- **/EPF/Templates:** todos los archivos contenidos en esta carpeta permiten la completa gestión de las plantillas asociadas a los productos de trabajo utilizadas por FVE. Hay ficheros que permiten la creación de plantillas, otros su edición, borrado o visualización.
- **/EPF/Workproducts:** contiene todos los archivos necesarios para hacer funcionar la gestión de productos de trabajo. Se permite la creación, borrado, edición y visualización de productos de trabajo.

Una pregunta que puede surgir llegado a este punto es cómo hacer encajar todas las piezas que tenemos sobre la mesa para que la aplicación funcione. Nosotros aportamos una serie de archivos pero, ¿en qué momento son utilizados? De ello hablaremos en el siguiente apartado.

Funcionamiento de la aplicación

Siguiendo los convenios de agregación de módulos creados por dotProject, nosotros debemos implementar un módulo con los datos y archivos que queramos. DotProject presentará de todo este nuevo contenido cuando los usuarios accedan de modo directo a nuestro módulo, pulsando el enlace que aparece en la parte superior de la pantalla:



Menú superior de aplicación.

Tras acceder de esta forma a nuestro módulo, se mostrará por pantalla el contenido del archivo index.php ubicado en la raíz de la carpeta que contiene nuestro módulo.

Sin embargo, es posible también realizar accesos a nuestro módulo aunque éste no se encuentre visible en la barra superior, sólo hay que conocer la ruta de los archivos y realizar las llamadas adecuadas.

Cuando accedemos a una página de dotProject, lo hacemos a través de una sentencia similar a la siguiente:

<http://localhost/FVE/index.php?m=fve&pag=mostrarWorkProduct&id=21>

Como se puede ver, existe paso de variables utilizando el método "GET" que ofrece HTML. A continuación analizamos la sentencia de llamada con las variables que se utilizan:

- **http://localhost/FVE/index.php**: esta sentencia llama al índice principal de dotProject. Aparece en cualquier página que utilicemos ya que este índice es el general de toda la aplicación y siempre se pasa por él.
- **m=fve**: este es el selector de módulos que utiliza dotProject. Nuestro módulo tiene como código interno 'fve', así que dotProject lo que hará será direccionar la página pedida por el usuario al index.php que se encuentre dentro de la carpeta del módulo. En nuestro caso, a FVE/index.php.
- **pag=mostrarWorkProduct**: esta variable es ya de uso interno en nuestro módulo. Lo que hará será seleccionar una funcionalidad concreta dentro de nuestro módulo. En este caso irá a la página de mostrar un producto de trabajo.
- **id=21**: de nuevo, se trata de una variable de uso interno en nuestro módulo. En este caso, lo que hace esta variable es identificar el producto de trabajo que debe mostrarse.

En la siguiente imagen se ilustra de manera visual mediante un diagrama de secuencia las llamadas que se llevan a cabo entre los distintos archivos de la aplicación para servir al usuario la página que éste demanda. El proceso es análogo para cualquier página que se pida dentro del módulo FVE.

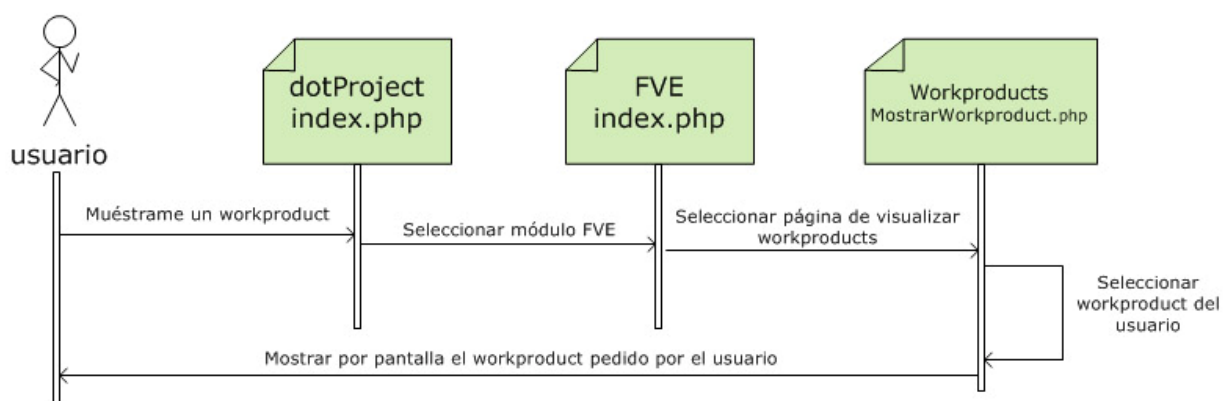


Diagrama de secuencia que explica cómo se llevan a cabo las llamadas entre los archivos para responder a una petición de una página web en FVE Project Manager por parte del usuario.

En resumen, dentro del archivo *index.php* que se encuentra en la raíz de la carpeta de nuestro módulo se encuentra una estructura 'switch' de tamaño considerable que se encarga de filtrar lo que llega por la variable '*pag*' e

incluye en el contenido de la página el archivo con extensión php que se corresponda con el valor de la variable. A continuación se muestra una fracción de este código:

```
switch ($_GET[pag]) {
    case "intro": include "intro.php";
    break;
    case "proyectos": include "proyectos.php";
    break;
    case "usuarios": include "users/usuarios.php";
    break;
    case "verUsuario": include "users/verUsuario.php";
    break;
    case "listarWorkProduct": include "epf/workproducts/list.php";
    break;
    case "crearWorkProduct": include "epf/workproducts/create.php";
    break;
    ...
}
```

Código relativo a clase index.php.

Como se puede apreciar, lo único que hace el archivo de índice es direccionar desde él hasta la página que solicite el usuario. Este archivo de índice además coloca un menú en la parte superior de la pantalla mediante el cual se visualiza (siempre que pasemos por el index.php de FVE) diversos enlaces para acceder a páginas de ayuda que ofrece nuestro módulo.



Submenú de la aplicación.

De cara a los archivos que se modifican sobre dotProject (se explican en detalle en el apartado 4.1.3), es necesario explicar que lo único que realizan estos enlaces es una llamada al index.php de nuestro módulo mediante llamadas como las que aparecen anteriormente. Por ejemplo, al pulsar sobre el botón de edición de un rol que aparezca en la pestaña "EPF Roles" de la pantalla de información de nuestro proyecto, se realizará una redirección a la siguiente página:

<http://localhost/FVE/index.php?m=fve&pag=editarRol&role=33>

Sobre esta URL, nuestro archivo index.php redireccionará a la página /FVE/epf/roles/setRoleInfo.php.

4.3.2.2 Clases utilizadas

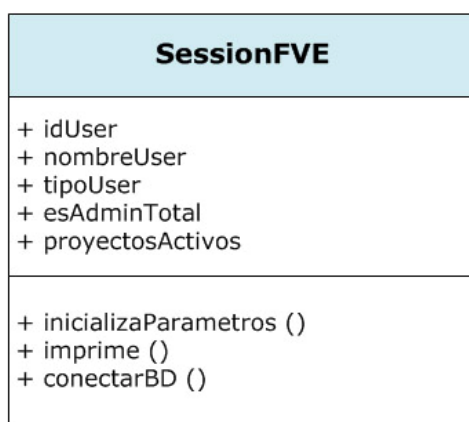
La utilización de clases en nuestro sistema tiene fundamentalmente dos objetivos: almacenar información de la sesión del usuario y facilitar el acceso a los datos de las tablas de la base de datos. Todos los archivos de gestión de clases se encuentran ubicados dentro de la carpeta /FVE/Clases.

Seguidamente se especificarán más en detalle cada una de las citadas características.

Sesiones

En nuestra aplicación se hace uso de dos sesiones diferentes. Estas sesiones funcionan de manera independiente y no interfieren en el sistema de sesiones que utiliza dotProject.

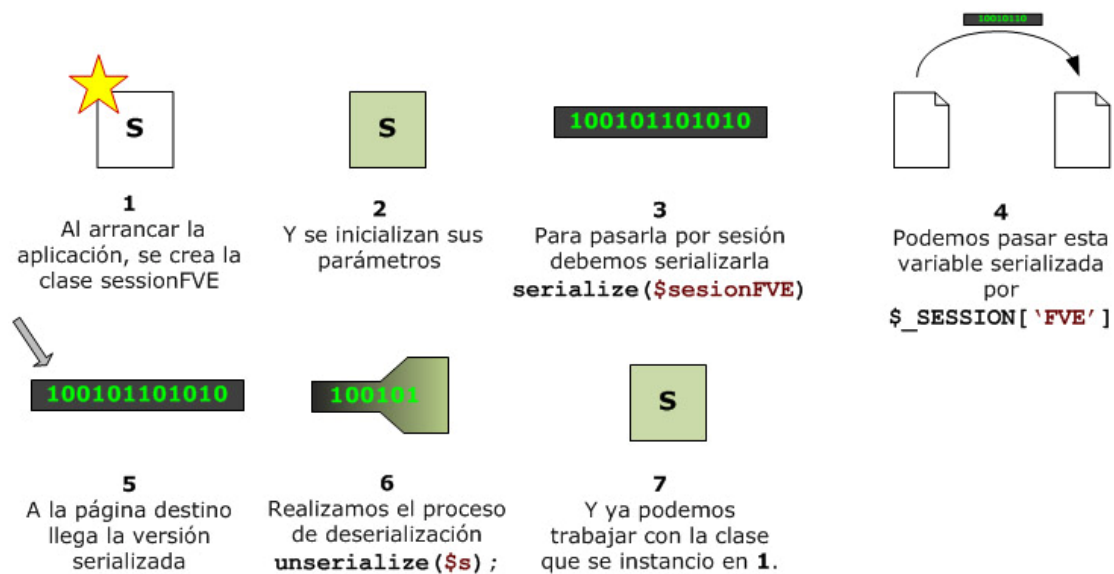
La primera de estas sesiones es la **Sesión FVE**. Se encarga de controlar las variables globales que se utilizan en nuestro módulo. La estructura con la información principal de esta clase es similar a la que aparece en la siguiente imagen:



Clases SessionFVE.

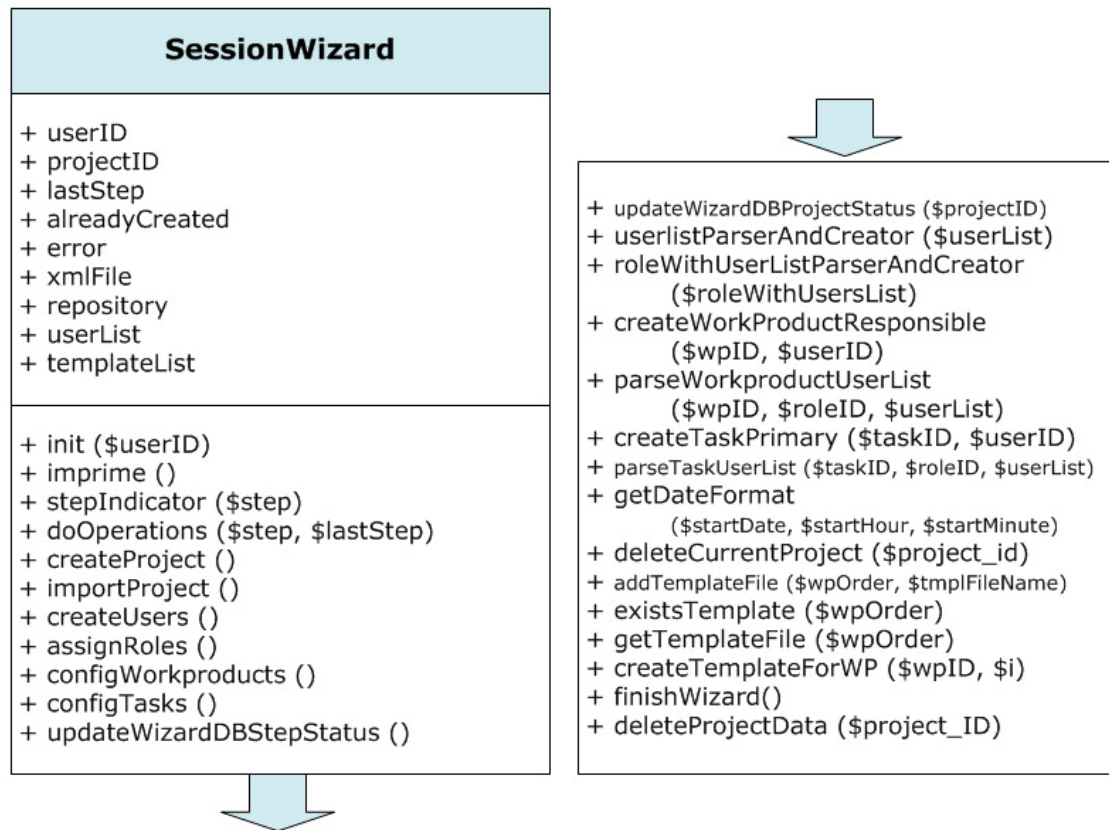
Esta clase se instancia al arrancar la aplicación, donde se hace uso del método `inicializaParametros ()`. En base al usuario que esté iniciado en la aplicación, así se inicializarán las variables. `imprime ()` se utiliza para tareas de depuración e imprime por pantalla los valores de los atributos de la clase. `conectarBD` lo utiliza la misma clase para acceder a la base de datos utilizada.

La forma de mantener esta clase siempre accesible desde cualquier punto de la aplicación consiste en almacenarla en los atributos de sesión de HTML. El problema que encontramos fue el de que una clase no se puede pasar directamente como atributo de sesión, es preciso realizar previamente una *serialización* de la misma. Cuando se recupera la variable en la página destino, es preciso realizar el proceso inverso antes de poder recuperar información de la clase mediante un método de *deserialización*. En el siguiente gráfico se muestra el proceso llevado a cabo:



Secuencia de acciones en la sesión FVE.

Por otro lado, tenemos la **Sesión del Wizard**. Esta sesión se crea únicamente cuando usuario accede al asistente de creación de proyectos. Sirve para controlar el proceso de creación (por qué paso va) así como variables auxiliares que ayudan a facilitar el proceso de creación del proyecto. Se ha optado por delegar toda la mayor parte de las funciones de creación de proyectos dentro de esta clase. Por ello, como se podrá apreciar en la siguiente imagen, esta clase tiene muchos métodos y atributos, pero ellos le permiten llevar a cabo una operación tan compleja como es instanciar y configurar un nuevo proyecto a partir de un archivo XML que contenga su definición.



Clases SessionWizard.

La página que controla el wizard es /FVE/epf/projects/wizard.php. Esta página se encarga de cargar la sesión, comprobar el paso en el que se encuentra y mostrar la información por pantalla con la información acorde al paso en el que se encuentre el usuario.

Esta clase necesita del mismo método de transferencia entre páginas (mediante el uso de sesiones de HTML) que se utilizaba en la sesión FVE.

Acceso a base de datos.

De acuerdo a los estándares en cada vez más entornos de programación, se han creado clases que implementan el acceso a cada tabla de la base de datos. De este modo, se pretende realizar sobre PHP una simplificación de los modelos existentes en otras tecnologías web como Hibernate (J2EE) o Active Record (Ruby On Rails) para convertir en objetos las entidades de las bases de datos. En nuestro caso estas clases cumplen la misión de mantener la independencia entre las operaciones con la base de datos y el resto de operaciones, además de favorecer la modularidad y la reutilización del código, evitando los clásicos y frecuentes inconvenientes de PHP en la identificación de espacios de nombres.

En nuestro sistema, hemos creado un total de 21 clases que nos permiten el acceso a 21 tablas distintas contenidas en la base de datos. En el siguiente

gráfico se mostrarán las características de estas clases y qué métodos fundamentales poseen. La nomenclatura empleada consiste en que el nombre de la clase hace referencia a la tabla a la que se accede. Así, por ejemplo, la clase `fveRolesAccess` accede a la tabla `fve_roles`.

fveParametersAccess + createParam (\$param_name,param_value) +deleteParam(\$param_name) +updateParam (\$param_name,\$param_value) +getParamValue (\$param_name)	fveProjectAccess + createProject (\$projectID, \$repositoryPath, \$guidelineLink) +deleteProject(\$projectID) +updateProject (\$projectID, \$repositoryPath, \$guidelineLink) +searchProjectIdInFveProjects (\$project_id) + getProject(\$project_id) +updateRepositoryPath (\$project_id, \$repositoryPath) +updatePluginProject (\$project_id,\$plugin,\$plugin_ desc,\$plugin_authors)	fveTasksAccess +getFveTaskWithId(\$task_id) +insertOnlyFveTask (\$task_id,\$task_epf_type) +updateOnlyFveTask (\$task_id,\$task_epf_type) +insertOnlyFveTaskWithId (\$task_id) + selectTasksFromProject (\$projectID) + deleteTasksFromProject (\$project_id) + getAllUserTasksInProject (\$user_id,\$project_id) +getAllUserPrimaryTasksInProjec (\$user_id,\$project_id) +getAllUserAdditionalTasksInProj (\$user_id,\$project_id) + getTasksWithWorkproduct (\$workproduct_id) + existeFveTask(\$task_id) +parseAdditionalRoles(\$roleList) + parseWorkproducts(\$tasksList) + parseTypes(\$typesList) + insertPrimaryRole (\$task_id,\$selectedRolePrimary) + insertAdditionalRoles (\$task_id,\$arrayRoles) + insertTasksRelations (\$task_id,\$arrayWorkproducts ,\$arrayTypes) + insertAdditionalRole (\$role_epf_id,\$task_id) + insertTaskAndWorkproducts (\$task_id,\$workproduct_id,\$d ependency_type) +deleteAdditionalRoles(\$task_id) +deletePrimaryRole(\$task_id) +deleteWorkproductsRelation sWithTask (\$task_id)
fveProjectAndWorkersAccess + createProjectWithUser (\$projectID, \$userID) +deleteAllUsersOfProject (\$projectID) +insertUsersInThisProject (\$project_id,\$arrayUsers) +selectUsersFromProject (\$project_id)	fveRoleModifiesWorkproductAccess + getRolesModifies (\$workproduct_id) + getWorkproductsModifies (\$role_epf_id) + getAdditionalRolesOfThis- Workproduct(\$workproduct_id)	
fveRoleAdditionalTaskAccess + getRolesAdditional (\$task_id) +getTasksAdditional (\$role_epf_id)	fveRoleResponsibleAccess +getRoleResponsible (\$workproduct_id) +getWorkproductsResponsible (\$role_epf_id)	

Clases de acceso a base de datos 1.

fveTemplatesAccess	fveRolePrimaryTaskAccess	fveRolesAccess
+insertTemplate (\$template_name, \$template_desc, \$template_path, \$epf_id, \$project_id) +insertTemplateWithId (\$template_name, \$template_desc, \$template_path, \$epf_id, \$project_id, \$idTemplate) + selectAllTemplates() + selectTemplatesFromProject (\$projectId) + getTemplate(\$template_id) + updateTemplatePath (\$project_id,\$template_path, \$template_final_path) + deleteTemplateAndRelations (\$template_id) +deleteTemplateAndRelationsBy WorkproductId (\$workproductId)	+ getRolePrimary(\$task_id) +getTasksPrimary (\$role_epf_id)	+insertRole (\$role_name,\$role_desc,\$epf_id, \$project_id) +deleteRolesFromProject (\$project_id) +updateRole (\$role_epf_id,\$role_name,\$ro le_desc,\$epf_id) +getMaxId_fve_roles() +selectAllRoles() +selectRolesFromProject (\$projectId) + getDistinctSpecificRolesUser- InProject(\$user_id,\$project_id) + getRole(\$role_epf_id) + getRoleWithId (\$role_epf_id) + deleteRoleAndRelations (\$role_epf_id)
	fveUserAdditionalTaskAccess	
	+createAdditionalUser (\$user_id, \$role_epf_id, \$task_id) +deleteTaskRows (\$task_id, \$role_epf_id) + deleteModifiersWithTask (\$task_id)	

Clases de acceso a base de datos 2.

fveUserModifiesWorkproductAccess	fveUserPossibleRolesAccess	fveUserPrimaryTasksAccess
+ createModifierUser (\$user_id, \$role_epf_id, \$workproduct_id) +deleteWorkproductRows (\$workproduct_id, \$role_epf_id) deleteModifiersWithWorkproduct (\$workproduct_id) +getUserAndRoleModifies (\$workproduct_id)	+ createUserWithRole (\$role_epf_id, \$user_id) +deleteAllUsersOfRole(\$roleID) +getAllPossiblRolOfUserInProject (\$user_id,\$project_id) +getUsersThatAssumeThisRole (\$role_epf_id) + insertUsersInThisRole (\$role_epf_id,\$arrayUsers)	+ createPrimaryUser (\$user_id, \$task_id) +deleteTasksRows (\$task_id)

Clases de acceso a base de datos 3.

fveUsrResponsibleWorkproductAccess + createResponsibleUser (\$user_id, \$workproduct_id) + deleteWorkproductRows (\$workproduct_id)	fveWorkproductAndTemplatesAccess + insertWorkproductWithTemplat (\$wpID, \$templateID) + deleteRowsFromWorkproduct (\$wpID)	fveWorkproductsAccess + selectWorkproductsFromProject (\$projectID) + getTemplateAssociatedToWorkproduct (\$workproduct_id) + selectWorkproductsFromTask (\$taskID) + getAllUserWorkproductsModifierInProject (\$user_id,\$project_id) + getAllUserWorkproductsResponsibleInProject (\$user_id,\$project_id) + getAllUsrWorkproductsInProject (\$user_id,\$project_id) + insertArtifact (\$workproduct_name,\$workproduct_desc,\$epf_id,\$project_id) + getMaxId_fve_workproducts() + getName(\$workproduct_id) + getWorkproduct (\$workproduct_id) + deleteWorkproductAndRelations (\$workproduct_id) + editWorkproductGeneralInfo (\$workproduct_id,\$workproduct_name,\$workproduct_desc) + editWrkproductStatusInformation (\$workproduct_id,\$percentStatus,\$userStatus) + selectWorkproductsFromProjectWithPercentStatusRange (\$projectID,\$percentStatusLow,\$percentStatusHigh)
projectAccess + createProject (\$name, \$companyID, \$ownerID, \$startDate, \$endDate, \$description) + getProjectName(\$project_id) + deleteProject(\$project_id) + getProject(\$project_id)	tasksAccess + getTaskWithId(\$task_id) + editTaskStartDate (\$taskID, \$date) + editTaskEndDate (\$taskID, \$date) + fixTasksParent(\$projectID)	
userAccess + getUserName(\$user) + getUserPass(\$user)	userTasksAccess + insertRow(\$userID, \$taskID) + deleteRow(\$userID, \$taskID)	

Clases de acceso a base de datos 4.

4.4. Estructura de BBDD

4.4.1. Dependencia de dotProject

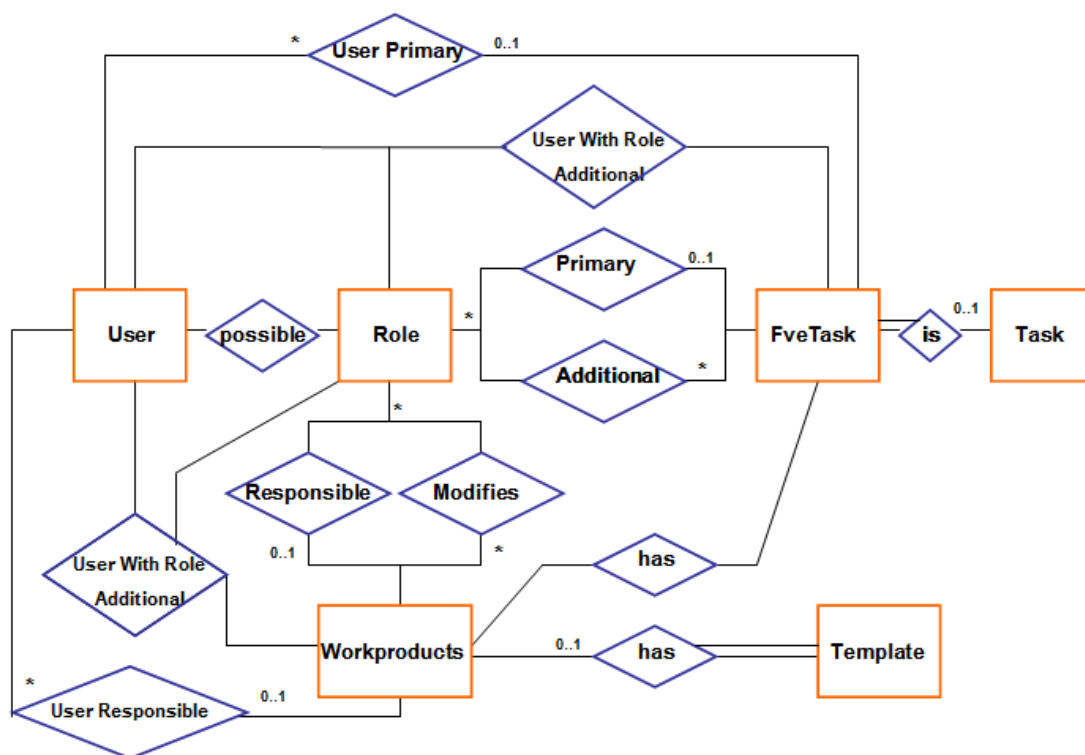
Los contenidos definidos en la aplicación deben ser almacenados en una base de datos para cumplir los requisitos de acceso rápido a los datos y de relación con los datos de dotProject. Debido a implementarse sobre un gestor de proyectos como dotProject no podía realizarse un diseño de base de datos independiente de este gestor, por lo que en el diseño se tuvieron en cuenta su estructura y principalmente qué tablas de las ya existentes sería relevantes en nuestra aplicación. Estas tablas, projects, tasks y users, serán explicadas posteriormente especificando qué mecanismos se han seguido para asegurar la compatibilidad.

4.4.2. Esquema Entidad – Relación

Debido al número de tablas y relaciones, se ha dividido el esquema entidad – relación realizado durante el diseño de la base de datos en 2

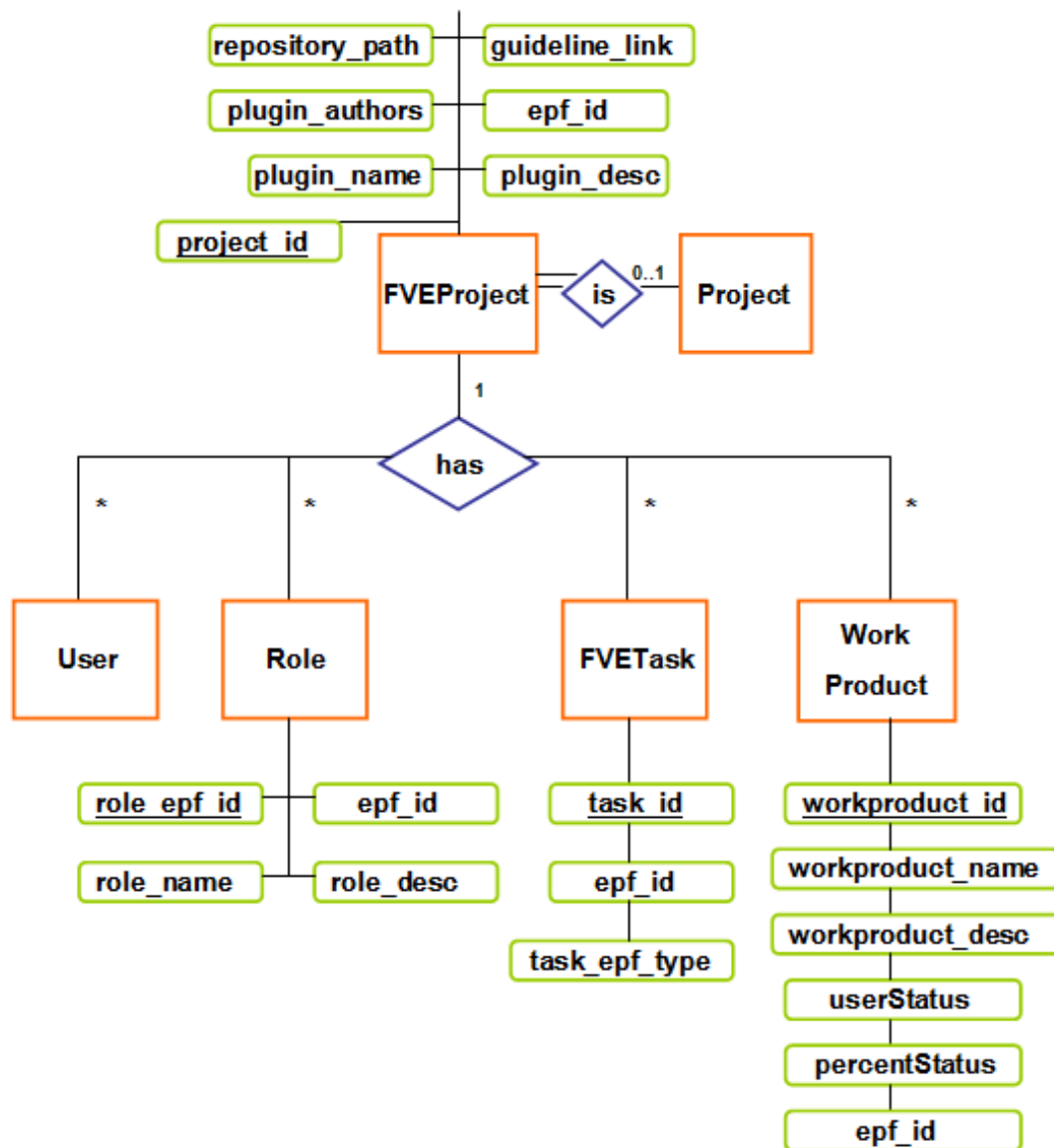
partes, y en la primera de ellas no se han incluido los campos de cada tabla. No obstante, los atributos de todas las tablas que finalmente forman parte del proyecto se muestran en el apartado 4.4.4.

- **Primer diagrama:** Muestra las entidades y propiedades necesarias para almacenar toda la información que se quiere importar de los procesos de desarrollo creados en EPF Composer.



Esquema Entidad – Relación 1.

- **Segundo diagrama:** Muestra la relación de las nuevas entidades con los proyectos y los campos de éstas (User y Project figuran en el esquema ya que han sido tenidas en cuenta para asegurar la compatibilidad con la base de datos de dotProject, pero no se añaden los campos porque estos son los que están definidos por defecto).



Esquema Entidad – Relación 2.

4.4.3. Explicación de la estructura

Tras evaluar la información a importar se crearon instancias sobre los elementos de EPF Composer que tendrían aplicación en nuestra aplicación: Roles, Tareas (diferenciadas como FVETasks), Workproducts y Templates. Las relaciones que se creaban entre ellas en EPF han sido traducidas como funciones, que delimitan la cardinalidad en lo referente a roles responsables o primarios (de productos de trabajo o tareas respectivamente).

Además, es necesario incluir información que relacione a los usuarios (información de dotProject) con los roles (información de EPF), de un modo que no está delimitado en ninguna aplicación. Para ello se han creado tablas alternativas, que tienen la ventaja de separar los datos teóricos sobre roles y otras entidades, con los usuarios que asumen ese rol. Esta separación añade una operación más en las consultas, pero evita la redundancia

inaceptable que se produciría si sobre cada instancia de esta relación se almacenarían todos los datos generales. Además, hay que destacar que la posibilidad de que un usuario pueda participar como adicional en una tarea o modificador de un workproduct, siendo varios los roles adicionales/modificadores introducidos, obliga a que haya una tabla ternaria por tarea y por workproducts para conocer con que rol desempeña su trabajo cada usuario en cada entidad. Todas estas puntualizaciones tienen también su correspondencia en el uso de las claves primarias que se hará posteriormente.

La estructura que muestran los esquemas entidad relación proporcionados se respetó en su mayoría en el paso a un modelo relacional que pudiera ser implementado en una base de datos SQL. Únicamente hay que notar que las relaciones totales entre tablas de dotProject y tablas propias (proyectos y tareas) no se han traducido como nuevas tablas, ya que al compartir id esta asociación no es necesaria, y que la pertenencia de un elemento a un proyecto (excepto los usuarios) se ha incluido como atributo de la entidad en vez de en una tabla que las relacionara, debido a los requisitos impuestos de que un rol, un producto de trabajo o una tarea sólo pueden pertenecer a un proyecto.

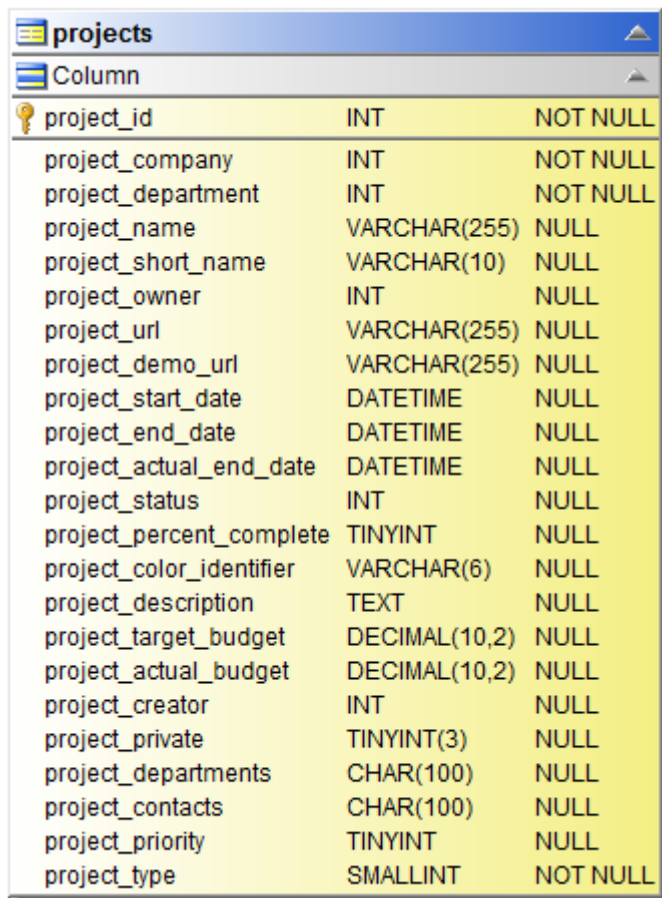
4.4.4. Listado de las tablas utilizadas

Esta versión de base de datos recoge la influencia de dos versiones anteriores que fueron descartadas para mejorar el manejo de los datos (sobre todo los importados de EPF Composer) y clarificar las relaciones entre las tablas (tanto específicas del módulo como genéricas de dotProject).

4.4.4.1. - Tablas que se utilizan de dotProject

4.4.4.1.1. - projects

Función: Es la tabla que almacena información sobre los proyectos definidos en dotProject. Las instancias de esta tabla se visualizarán en el módulo projects y podrán asociarse con tareas y compañías. Tiene información muy variada sobre el proyecto, con ciertos campos que podrían reutilizarse para introducir información del módulo, aunque resulta insuficiente para almacenar datos sobre el repositorio, descripción del plugin de EPF que se usa en la importación del proyecto, etc.

Estructura:


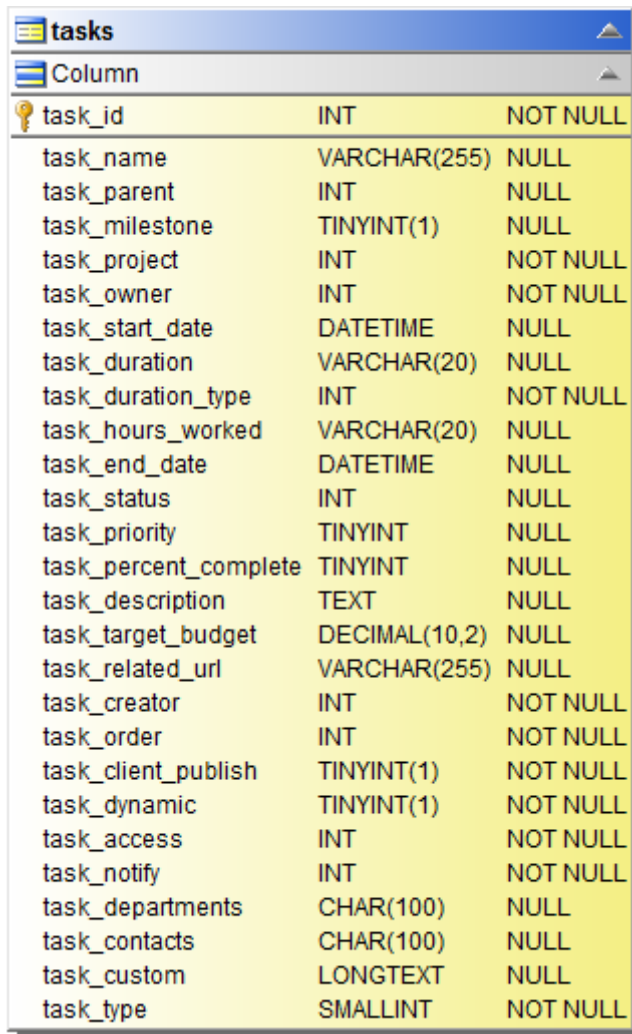
The image shows a screenshot of a database management tool window titled 'projects'. It displays the column structure for a table named 'projects'. The columns are listed with their data types and nullability constraints. The first column, 'project_id', is marked as the primary key with a key icon.

Column	INT	NOT NULL
project_id	INT	NOT NULL
project_company	INT	NOT NULL
project_department	INT	NOT NULL
project_name	VARCHAR(255)	NULL
project_short_name	VARCHAR(10)	NULL
project_owner	INT	NULL
project_url	VARCHAR(255)	NULL
project_demo_url	VARCHAR(255)	NULL
project_start_date	DATETIME	NULL
project_end_date	DATETIME	NULL
project_actual_end_date	DATETIME	NULL
project_status	INT	NULL
project_percent_complete	TINYINT	NULL
project_color_identifier	VARCHAR(6)	NULL
project_description	TEXT	NULL
project_target_budget	DECIMAL(10,2)	NULL
project_actual_budget	DECIMAL(10,2)	NULL
project_creator	INT	NULL
project_private	TINYINT(3)	NULL
project_departments	CHAR(100)	NULL
project_contacts	CHAR(100)	NULL
project_priority	TINYINT	NULL
project_type	SMALLINT	NOT NULL

Representación visual de tabla projects.

4.4.4.1.2. - tasks

Función: Es la tabla que almacena información sobre las tareas definidas en dotProject. Las instancias de esta tabla se representarán en casi todos los módulos de la aplicación, y solo por figurar en esta tabla soportan las funcionalidades de dotProject como diagramas de Gantt, informes, dependencias, etc. Aunque no existen campos para almacenar la nueva información, todas las tareas que se creen en nuestro módulo deben tener su información complementaria en esta tabla para asegurar la compatibilidad.

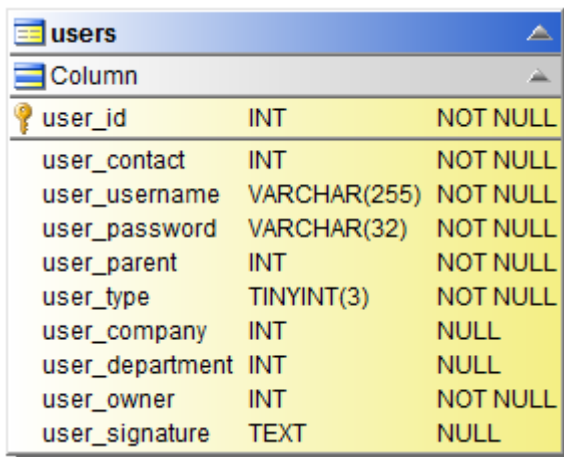
Estructura:


The image shows a screenshot of a database management tool window titled 'tasks'. Below the title bar is a 'Column' header. The table structure is displayed as follows:

Column	DataType	Nullable
task_id	INT	NOT NULL
task_name	VARCHAR(255)	NULL
task_parent	INT	NULL
task_milestone	TINYINT(1)	NULL
task_project	INT	NOT NULL
task_owner	INT	NOT NULL
task_start_date	DATETIME	NULL
task_duration	VARCHAR(20)	NULL
task_duration_type	INT	NOT NULL
task_hours_worked	VARCHAR(20)	NULL
task_end_date	DATETIME	NULL
task_status	INT	NULL
task_priority	TINYINT	NULL
task_percent_complete	TINYINT	NULL
task_description	TEXT	NULL
task_target_budget	DECIMAL(10,2)	NULL
task_related_url	VARCHAR(255)	NULL
task_creator	INT	NOT NULL
task_order	INT	NOT NULL
task_client_publish	TINYINT(1)	NOT NULL
task_dynamic	TINYINT(1)	NOT NULL
task_access	INT	NOT NULL
task_notify	INT	NOT NULL
task_departments	CHAR(100)	NULL
task_contacts	CHAR(100)	NULL
task_custom	LONGTEXT	NULL
task_type	SMALLINT	NOT NULL

Representación visual de tabla tasks.**4.4.4.1.3. - users**

Función: Es la tabla que almacena información sobre los usuarios existentes en dotProject. Las instancias de esta tabla permiten que un usuario pueda asumir una tarea, adquirir permisos específicos, y obtener una visión particularizada del proyecto. Debido a los objetivos de nuestro proyecto, trabajar con esta tabla será indispensable para asegurar la consistencia entre los usuarios de dotProject y Subversion y para permitir trabajar con los contenidos importados de EPF Composer.

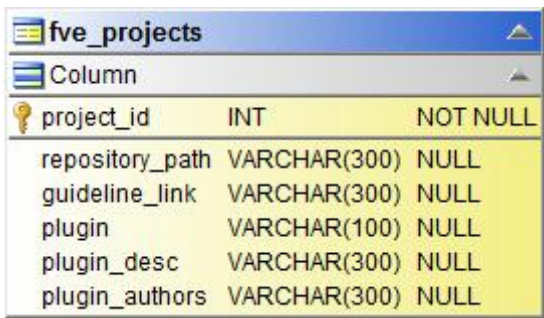
Estructura:


users		
Column		
user_id	INT	NOT NULL
user_contact	INT	NOT NULL
user_username	VARCHAR(255)	NOT NULL
user_password	VARCHAR(32)	NOT NULL
user_parent	INT	NOT NULL
user_type	TINYINT(3)	NOT NULL
user_company	INT	NULL
user_department	INT	NULL
user_owner	INT	NOT NULL
user_signature	TEXT	NULL

Representación visual de tabla users.

4.3.4.2: Tablas nuevas**4.4.4.2.1. - fve_projects**

Función: Incorporar a los proyectos de dotProject una serie de atributos propios de nuestro módulo FVE. Estos proyectos tendrán como identificador el correspondiente al project_id de dotProject. Incluye información sobre el plugin que solo será rellenada si un proyecto se crea a partir de un fichero XML de EPF Composer.

Estructura:


fve_projects		
Column		
project_id	INT	NOT NULL
repository_path	VARCHAR(300)	NULL
guideline_link	VARCHAR(300)	NULL
plugin	VARCHAR(100)	NULL
plugin_desc	VARCHAR(300)	NULL
plugin_authors	VARCHAR(300)	NULL

Representación visual de tabla fve_projects.

Script SQL de creación:

```
CREATE TABLE fve_projects (
    project_id INT,
    repository_path VARCHAR(300),
    guideline_link VARCHAR(300),
    plugin VARCHAR(100),
    plugin_desc VARCHAR(300),
    plugin_authors VARCHAR(300),
    PRIMARY KEY (project_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id)
```

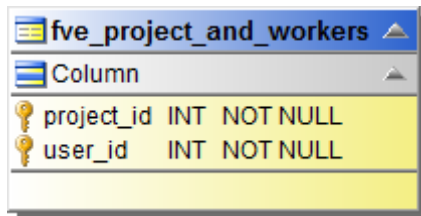
```
) type=MyISAM;
```

Observaciones:

4.4.4.2.2. - fve_project_and_workers

Función: Especifica qué usuarios podrán trabajar en un proyecto (aunque en un momento determinado no tengan tareas asignadas en el proyecto).

Estructura:



fve_project_and_workers	
Column	
project_id	INT NOT NULL
user_id	INT NOT NULL

Representación visual de tabla fve_project_and_workers.

Script SQL de creación:

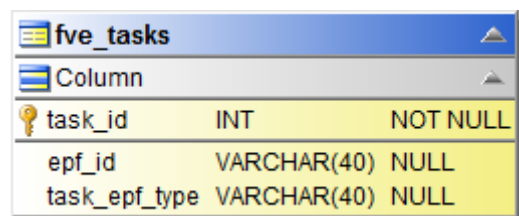
```
CREATE TABLE fve_project_and_workers (
    project_id      INT,
    user_id         INT,
    PRIMARY KEY (project_id,user_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
) type=MyISAM;
```

Observaciones:

4.4.4.2.3. - fve_tasks

Función: Especifica información adicional de la tarea (qué epf_id y qué tipo tiene la tarea de acuerdo a la importación de EPF), y se relaciona con la tabla tasks de dotProject por tener el mismo identificador único.

Estructura:



fve_tasks	
Column	
task_id	INT NOT NULL
epf_id	VARCHAR(40) NULL
task_epf_type	VARCHAR(40) NULL

Representación visual de tabla fve_tasks.

Script SQL de creación:

```
CREATE TABLE fve_tasks (
```



```

task_id          INT,
epf_id           VARCHAR(40),
task_epf_type    VARCHAR(40),
PRIMARY KEY (task_id),
FOREIGN KEY (task_id) REFERENCES tasks(task_id)
) type=MyISAM;

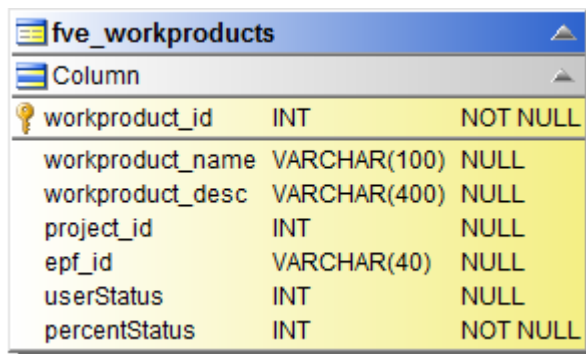
```

Observaciones:

4.4.4.2.4. - fve_workproducts

Función: Especifica qué productos de trabajo asociados a proyectos existen en la aplicación. Al utilizar identificadores únicos podrán existir dos workproducts "analistas" diferentes si pertenecen a distintos proyectos.

Estructura:



Column	Type	Constraints
workproduct_id	INT	NOT NULL, PRIMARY KEY
workproduct_name	VARCHAR(100)	NULL
workproduct_desc	VARCHAR(400)	NULL
project_id	INT	NULL, FOREIGN KEY
epf_id	VARCHAR(40)	NULL
userStatus	INT	NULL
percentStatus	INT	NOT NULL

Representación visual de tabla fve_workproducts.

Script SQL de creación:

```

CREATE TABLE fve_workproducts (
    workproduct_id INT,
    workproduct_name VARCHAR(100),
    workproduct_desc VARCHAR(400),
    project_id INT,
    epf_id VARCHAR(40),
    percentStatus INT,
    userStatus INT,
    PRIMARY KEY (workproduct_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id)
) type=MyISAM;

```

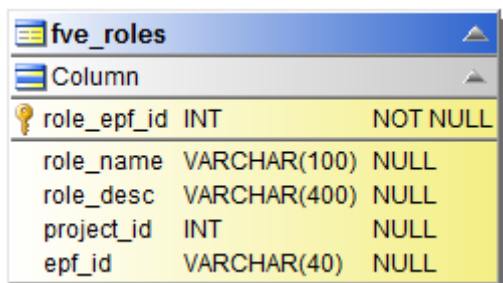
Observaciones:

- Existen campos del workproduct, sobre su estado, que son similares a los de las tareas de dotProject (como el porcentaje del workproduct), pero que deben añadirse aquí al no relacionarse con ninguna tabla de dotProject al contrario que en las tareas.

4.4.4.2.5. - fve_roles

Función: Especifica qué roles asociados a proyectos existen en la aplicación.

Estructura:



fve_roles		
Column		
role_epf_id	INT	NOT NULL
role_name	VARCHAR(100)	NULL
role_desc	VARCHAR(400)	NULL
project_id	INT	NULL
epf_id	VARCHAR(40)	NULL

Representación visual de tabla fve_roles.

Script SQL de creación:

```
CREATE TABLE fve_roles (
    role_epf_id INT,
    role_name VARCHAR(100),
    role_desc VARCHAR(400),
    project_id INT,
    epf_id VARCHAR(40),
    PRIMARY KEY (role_epf_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id)
) type=MyISAM;
```

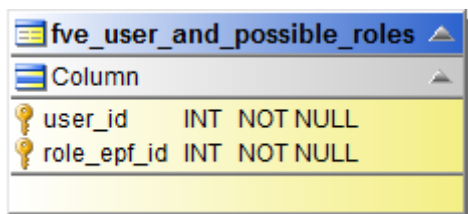
Observaciones:

- No hay relación con roles de dotProject por no presentar prácticamente funcionalidades en esos roles.

4.4.4.2.6. - fve_user_and_possible_roles

Función: Especifica qué roles de los especificados en un proyecto puede desempeñar (no desempeñar obligatoriamente) un usuario asignado a un proyecto.

Estructura:



fve_user_and_possible_roles		
Column		
user_id	INT	NOT NULL
role_epf_id	INT	NOT NULL

Representación visual de tabla fve_user_and_possible_roles.

Script SQL de creación:

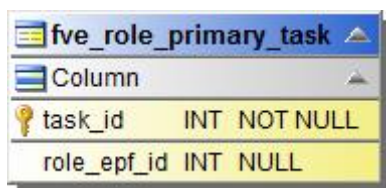
```
CREATE TABLE fve_user_and_possible_roles (
    role_epf_id    INT,
    user_id        INT,
    PRIMARY KEY (user_id, role_epf_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
) type=MyISAM;
```

Observaciones:

- No es necesario incluir el proyecto en la tabla, porque el proyecto está unívocamente determinado por el rol (es un campo de la tabla fve_roles, y un rol solo puede pertenecer un proyecto).

4.4.4.2.7. - fve_role_primary_task

Función: Almacena la información importada de EPF sobre el rol responsable del progreso de una tarea.

Estructura:


fve_role_primary_task	
Column	
task_id	INT NOT NULL
role_epf_id	INT NULL

Representación visual de tabla fve_role_primary_task.

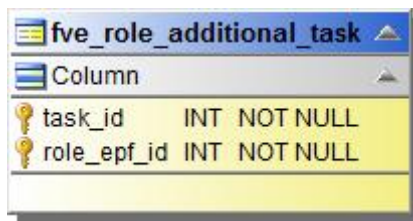
Script SQL de creación:

```
CREATE TABLE fve_role_primary_task (
    task_id        INT,
    role_epf_id    INT,
    PRIMARY KEY (task_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (task_id) REFERENCES tasks(task_id)
) type=MyISAM;
```

Observaciones: Solo hay un rol como responsable "primario" por tarea, por lo que la clave pasa a ser únicamente el atributo task_id.

4.4.4.2.8. - fve_role_additional_task

Función: Almacena la información importada de EPF sobre los roles que pueden participar en la modificación de una tarea.

Estructura:


fve_role_additional_task	
Column	
task_id	INT NOT NULL
role_epf_id	INT NOT NULL

Representación visual de tabla fve_role_additional_task.

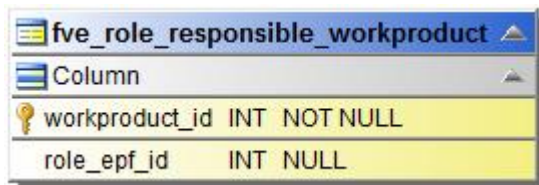
Script SQL de creación:

```
CREATE TABLE fve_role_additional_task (
    task_id          INT,
    role_epf_id      INT,
    PRIMARY KEY (task_id,role_epf_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (task_id) REFERENCES tasks(task_id)
) type=MyISAM;
```

Observaciones: Puede haber 0, 1 o más roles adicionales.

4.4.4.2.9. - fve_role_responsible_workproduct

Función: Almacena la información importada de EPF sobre el rol responsable del progreso de una tarea.

Estructura:


fve_role_responsible_workproduct	
Column	
workproduct_id	INT NOT NULL
role_epf_id	INT NULL

Representación visual de tabla fve_role_responsible_workproduct.

Script SQL de creación:

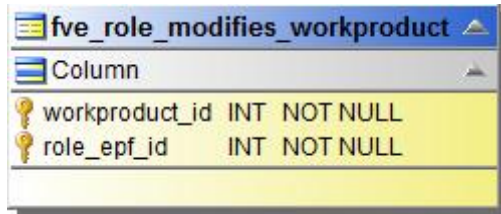
```
CREATE TABLE fve_role_responsible_workproduct (
    workproduct_id    INT,
    role_epf_id       INT,
    PRIMARY KEY (workproduct_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (workproduct_id) REFERENCES
fve_workproducts(workproduct_id)
) type=MyISAM;
```

Observaciones: Solo hay un rol responsable por workproduct, por lo que la clave pasa a ser únicamente el atributo workproduct_id.

4.4.4.2.10. - fve_role_modifies_workproduct

Función: Almacena la información importada de EPF sobre los roles que pueden participar en la modificación de un workproduct.

Estructura:



fve_role_modifies_workproduct	
Column	
workproduct_id	INT NOT NULL
role_epf_id	INT NOT NULL

Representación visual de tabla fve_role_modifies_workproduct.

Script SQL de creación:

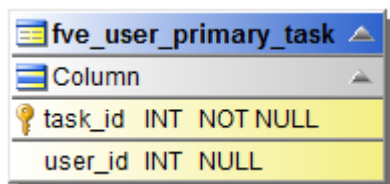
```
CREATE TABLE fve_role_modifies_workproduct (
    workproduct_id INT,
    role_epf_id INT,
    PRIMARY KEY (workproduct_id,role_epf_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (workproduct_id) REFERENCES
workproducts(workproduct_id)
) type=MyISAM;
```

Observaciones: Puede haber 0, 1 o más roles adicionales.

4.4.4.2.11. - fve_user_primary_task

Función: Almacena la información del usuario responsable "primario" de una tarea (ocupando el rol en que se basa la tabla fve_role_primary_task con el contenido importado de EPF).

Estructura:



fve_user_primary_task	
Column	
task_id	INT NOT NULL
user_id	INT NULL

Representación visual de tabla fve_user_primary_task.

Script SQL de creación:

```
CREATE TABLE fve_user_primary_task (
    task_id INT,
    user_id INT,
    PRIMARY KEY (task_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (task_id) REFERENCES tasks(task_id)
```

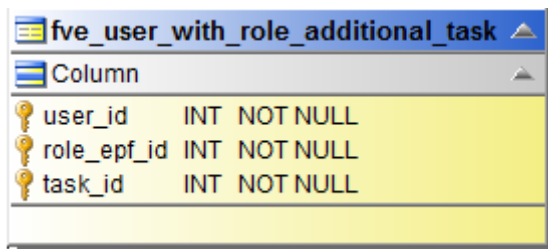
```
) type=MyISAM;
```

Observaciones: Solo hay un rol como responsable "primario" por tarea, por lo que la clave puede ser únicamente el atributo task_id.

4.4.4.2.12. - fve_user_with_role_additional_task

Función: Almacena la información del usuario adicional de una tarea (ocupando el rol en que se basa la tabla fve_role_additional_task con el contenido importado de EPF).

Estructura:



fve_user_with_role_additional_task	
Column	
user_id	INT NOT NULL
role_epf_id	INT NOT NULL
task_id	INT NOT NULL

Representación visual de tabla fve_user_with_role_additional_task.

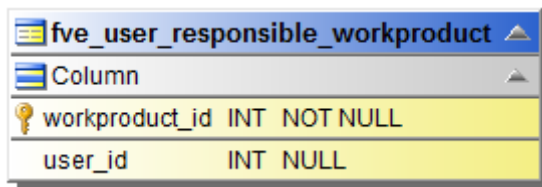
Script SQL de creación:

```
CREATE TABLE fve_user_with_role_additional_task (
    user_id          INT,
    role_epf_id      INT,
    task_id          INT,
    PRIMARY KEY (user_id,role_epf_id,task_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (task_id) REFERENCES tasks(task_id)
) type=MyISAM;
```

Observaciones: Es necesaria una tabla ternaria ya que puede existir más de un rol adicional, y con una tabla binaria en este lugar no podría asegurarse con qué rol es modificador un usuario en una tarea concreta.

4.4.4.2.13. - fve_user_responsible_workproduct

Función: Almacena la información del usuario responsable de un workproduct (ocupando el rol en que se basa la tabla fve_role_responsible_workproduct con el contenido importado de EPF).

Estructura:


fve_user_responsible_workproduct	
Column	
workproduct_id	INT NOT NULL
user_id	INT NULL

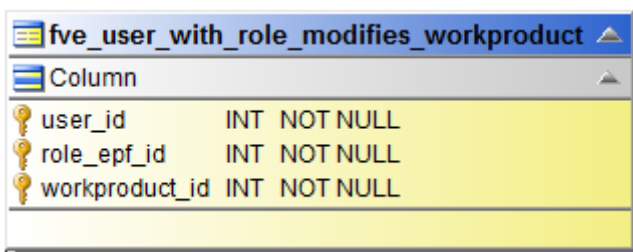
Representación visual de tabla fve_user_responsible_workproduct.

Script SQL de creación con atributos:

```
CREATE TABLE fve_user_responsible_workproduct (
    user_id          INT,
    workproduct_id INT,
    PRIMARY KEY (workproduct_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (workproduct_id) REFERENCES workproducts(workproduct_id)
) type=MyISAM;
```

Observaciones:**4.4.4.2.14. - fve_user_with_role_modifies_workproduct**

Función: Almacena la información del usuario que puede modificar un workproduct (ocupando uno de los roles en que se basa la tabla fve_role_modifies_workproduct con el contenido importado de EPF).

Estructura:


fve_user_with_role_modifies_workproduct	
Column	
user_id	INT NOT NULL
role_epf_id	INT NOT NULL
workproduct_id	INT NOT NULL

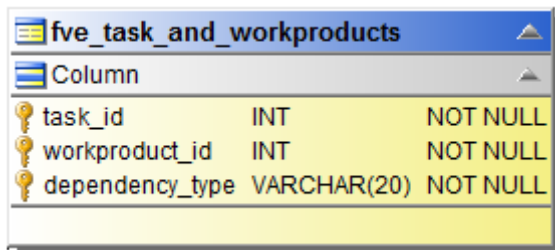
Representación visual de tabla fve_user_with_role_modifies_workproduct.

Script SQL de creación:

```
CREATE TABLE fve_user_with_role_modifies_workproduct (
    user_id          INT,
    role_epf_id      INT,
    workproduct_id INT,
    PRIMARY KEY (user_id,role_epf_id,workproduct_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (role_epf_id) REFERENCES fve_roles(role_epf_id),
    FOREIGN KEY (workproduct_id) REFERENCES workproducts(workproduct_id)
) type=MyISAM;
```

Observaciones:**4.4.4.2.15. - fve_task_and_workproducts**

Función: Relaciona las tareas con los workproducts, de modo que se indica qué tarea se ve afectada por qué workproduct y en qué medida, esto es, si es una entrada obligatoria, una entrada opcional o una salida de la tarea.

Estructura:


fve_task_and_workproducts			
Column			
task_id	INT	NOT NULL	
workproduct_id	INT	NOT NULL	
dependency_type	VARCHAR(20)	NOT NULL	

Representación visual de tabla fve_task_and_workproducts.

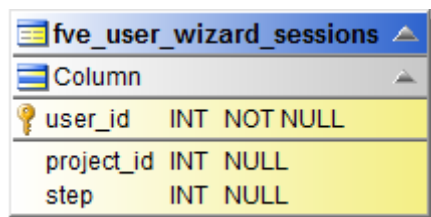
Script SQL de creación:

```
CREATE TABLE fve_task_and_workproducts (
    task_id          int,
    workproduct_id   int,
    dependency_type   varchar(20),
    PRIMARY KEY (task_id,workproduct_id,dependency_type),
    FOREIGN KEY      (workproduct_id) REFERENCES
fve_workproducts(workproduct_id),
    FOREIGN KEY (task_id) REFERENCES tasks(task_id)
) type=MyISAM;
```

Observaciones: dependency_type debe tener uno de estos valores: Mandatory input, Optional input o Output.

4.4.4.2.16. - fve_user_wizard_sessions

Función: Permite almacenar información de la sesión del wizard de instanciación de proyectos que el usuario dejó suspendida para su posterior reanudación.

Estructura:


fve_user_wizard_sessions			
Column			
user_id	INT	NOT NULL	
project_id	INT	NULL	
step	INT	NULL	

Representación visual de tabla fve_user_wizard_sessions.

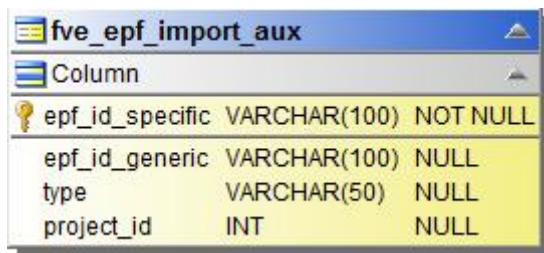
Script SQL de creación:

```
CREATE TABLE fve_user_wizard_sessions(
    user_id          INT,
    project_id       INT,
    step             INT,
    PRIMARY KEY (user_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id)
) type=MyISAM;
```

Observaciones: Es una función interna no modificable por el usuario.

4.4.4.2.17. - fve_epf_import_aux

Función: Se utiliza durante la fase de parseo del fichero XML con el que se importa el contenido de EPF, para posteriormente no almacenar más información. Son útiles para agilizar ciertas operaciones en el parseo. En concreto almacena el epf_id que tiene una tarea, rol, etc. como entidad general (epf_id_generic), y el epf_id diferente que recibe dicha tarea, rol, etc. cuando forma parte de un proceso de desarrollo (epf_id_specific).

Estructura:


Column	Type	Null
epf_id_specific	VARCHAR(100)	NOT NULL
epf_id_generic	VARCHAR(100)	NULL
type	VARCHAR(50)	NULL
project_id	INT	NULL

Representación visual de tabla fve_epf_import_aux.

Script SQL de creación:

```
CREATE TABLE fve_epf_import_aux (
    epf_id_generic VARCHAR(100),
    epf_id_specific VARCHAR(100),
    type          VARCHAR(50),
    project_id    INT,
    PRIMARY KEY (epf_id_specific)
) type=MyISAM;
```

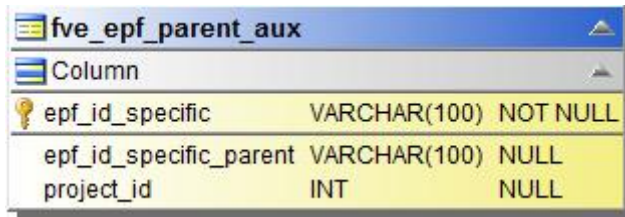
Observaciones: Es una función interna no modificable por el usuario.

4.4.4.2.18. - fve_epf_parent_aux

Función: Se utiliza durante la fase de parseo del fichero XML con el que se importa el contenido de EPF, para posteriormente no almacenar más información. Son útiles para agilizar ciertas operaciones en el parseo. En concreto almacena el epf_id específico que tiene una tarea como parte de un proceso, y el epf_id específico de la tarea padre (que en EPF será

generalmente de tipo task, activity, phase, iteration, deliveryProcess, capabilityPattern).

Estructura:



fve_epf_parent_aux		
Column		
epf_id_specific	VARCHAR(100)	NOT NULL
epf_id_specific_parent	VARCHAR(100)	NULL
project_id	INT	NULL

Representación visual de tabla fve_epf_parent_aux.

Script SQL de creación:

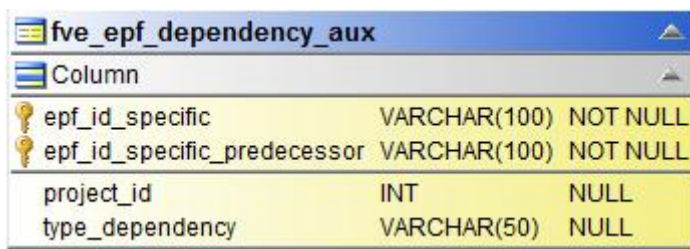
```
CREATE TABLE fve_epf_parent_aux (
    epf_id_specific VARCHAR(100),
    epf_id_specific_parent VARCHAR(100),
    project_id INT,
    PRIMARY KEY (epf_id_specific)
) type=MyISAM;
```

Observaciones: Es una función interna no modificable por el usuario.

4.4.4.2.19. - fve_epf_dependency_aux

Función: Se utiliza durante la fase de parseo del fichero XML con el que se importa el contenido de EPF, para posteriormente no almacenar más información. Son útiles para agilizar ciertas operaciones en el parseo. En concreto almacena el epf_id específico que tiene una tarea como parte de un proceso, y el epf_id específico de la tarea que la debe preceder, además de un atributo que almacenará el tipo de dependencia (finishToStart o finishToFinish en principio).

Estructura:



fve_epf_dependency_aux		
Column		
epf_id_specific	VARCHAR(100)	NOT NULL
epf_id_specific_predecessor	VARCHAR(100)	NOT NULL
project_id	INT	NULL
type_dependency	VARCHAR(50)	NULL

Representación visual de tabla fve_epf_dependency_aux

Script SQL de creación:

```
CREATE TABLE fve_epf_dependency_aux (
    epf_id_specific VARCHAR(100),
    epf_id_specific_predecessor VARCHAR(100),
    project_id INT,
```

```

type_dependency    VARCHAR(50),
PRIMARY KEY (epf_id_specific,epf_id_specific_predecessor)
) type=MyISAM;

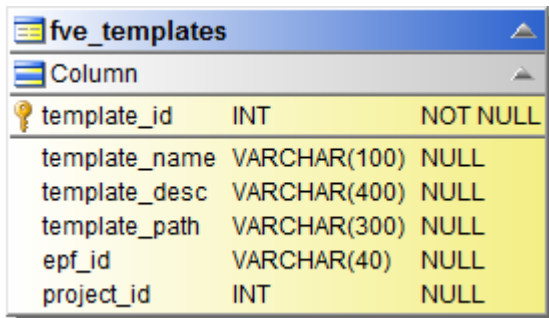
```


Observaciones: Es una función interna no modificable por el usuario.

4.4.4.2.20. - fve_templates

Función: Almacena la información de los Templates importados de EPF. Almacena en el campo `template_path` la ruta de las plantillas en el ordenador del usuario que se obtiene en la importación, aunque posteriormente se puede decidir dar un uso a este campo diferente y almacenar información de su localización en el servidor, o agregar otro campo para esta labor.

Estructura:



Column		
 template_id	INT	NOT NULL
template_name	VARCHAR(100)	NULL
template_desc	VARCHAR(400)	NULL
template_path	VARCHAR(300)	NULL
epf_id	VARCHAR(40)	NULL
project_id	INT	NULL

Representación visual de tabla `fve_templates`.

Script SQL de creación:

```

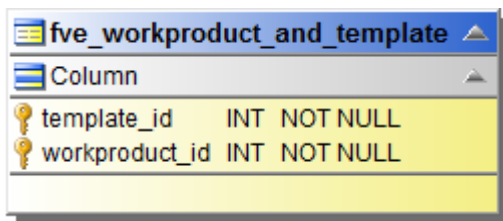
CREATE TABLE fve_templates (
    template_id    INT,
    template_name  VARCHAR(100),
    template_desc  VARCHAR(400),
    template_path  VARCHAR(300),
    epf_id         VARCHAR(40),
    project_id     INT,
    PRIMARY KEY (template_id),
    FOREIGN KEY (project_id) REFERENCES projects(project_id)
) type=MyISAM;

```

Observaciones:

4.4.4.2.21. - fve_workproduct_and_template

Función: Almacena la información de la plantilla que se utiliza para cada workproduct. En principio parece lógico que haya únicamente una plantilla por workproduct, pero se permite de momento por si es necesario que haya más de una en la base de datos, y sino posteriormente podría restringirse esta posibilidad mediante el cambio de la clave primaria de la tabla a únicamente `workproduct_id`.

Estructura:


fve_workproduct_and_template	
Column	
template_id	INT NOT NULL
workproduct_id	INT NOT NULL

Representación visual de tabla fve_workproduct_and_template.

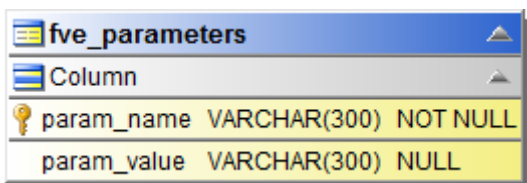
Script SQL de creación:

```
CREATE TABLE fve_workproduct_and_template (
    template_id INT,
    workproduct_id INT,
    PRIMARY KEY (template_id, workproduct_id),
    FOREIGN KEY (template_id) REFERENCES fve_templates(template_id)
    FOREIGN KEY (workproduct_id) REFERENCES fve_workproducts
    (workproduct_id)
) type=MyISAM;
```

Observaciones: Si se decidieran incluir otro tipo de guías de epf esta tabla debería convertirse en una tabla générica, pero con los requisitos actuales es suficiente.

4.4.4.2.22. - fve_parameters

Función: Almacena la información de configuración necesaria para FVE, lo que mejora la flexibilidad de la aplicación para ser instalado en diferentes entornos. Principalmente se almacenan datos de Subversion como el puerto utilizado o la dirección donde se encuentra el módulo j2ee encargado.

Estructura:


fve_parameters	
Column	
param_name	VARCHAR(300) NOT NULL
param_value	VARCHAR(300) NULL

Representación visual de tabla fve_parameters.

Script SQL de creación:

```
CREATE TABLE fve_parameters (
    param_name VARCHAR(300),
    param_value VARCHAR(300),
    PRIMARY KEY(param_name)
) type=MyISAM;
```

Observaciones: Se ha adoptado este nombre genérico de función y atributos de cara a posibles extensiones y a poder aumentar progresivamente el número de opciones configurables sin tener que crear nuevas tablas en la base de datos.

4.4.5 – Transaccionalidad en la base de datos

Se ha habilitado un sistema que permite transaccionalizar las operaciones que se realizan sobre la base de datos. De esta forma, se evitan conflictos derivados del acceso concurrente a la instancia de la base de datos [MySQLHispano 99].

La solución que se emplea es la utilización de cerrojos (*locks*) a nivel de tabla. Explicaremos el funcionamiento de estos cerrojos más detalladamente.

Cuando un hilo (ejecución concreta de la aplicación) quiere realizar una modificación sobre una tabla de la base de datos, previamente inicia una transacción. Para ello, realiza un bloqueo en la tabla que va a modificar. De esta forma y hasta que libere el cerrojo, cualquier otro hilo que intente acceder a esa tabla será bloqueado hasta la liberación del cerrojo. Una vez haya realizado la operación el hilo poseedor del cerrojo, lo liberará confirmando (*commit*) los cambios realizados. El segundo hilo que estaba en espera, tendrá ya acceso a la tabla de la base de datos y podrá visualizar los cambios que realizó el primero hilo. La siguiente imagen muestra las acciones que se producen cuando dos hilos de la aplicación intentan acceder simultáneamente a la misma tabla de la base de datos.



Esquema de transaccionalidad.

Apache genera un hilo para cada cliente que accede a la aplicación desde una ubicación distinta. De esta forma, si dos usuarios pretenden acceder a una misma tabla que está bloqueada por uno de ellos, la aplicación

entenderá que se trata de dos procesos independiente y bloqueará el paso al hilo que no tenga prioridad sobre la tabla (en este caso, el que llegara después).

La utilización de este sistema de control de acceso a la base de datos se utiliza en partes de la aplicación sensibles a modificaciones simultáneas, como por ejemplo el wizard.

4.5. Estructura del módulo de Subversion

La integración de Subversion en el sistema se realiza en dos partes claramente diferenciadas en su implementación. En la imagen inferior mostramos la interrelación de todos estos componentes:

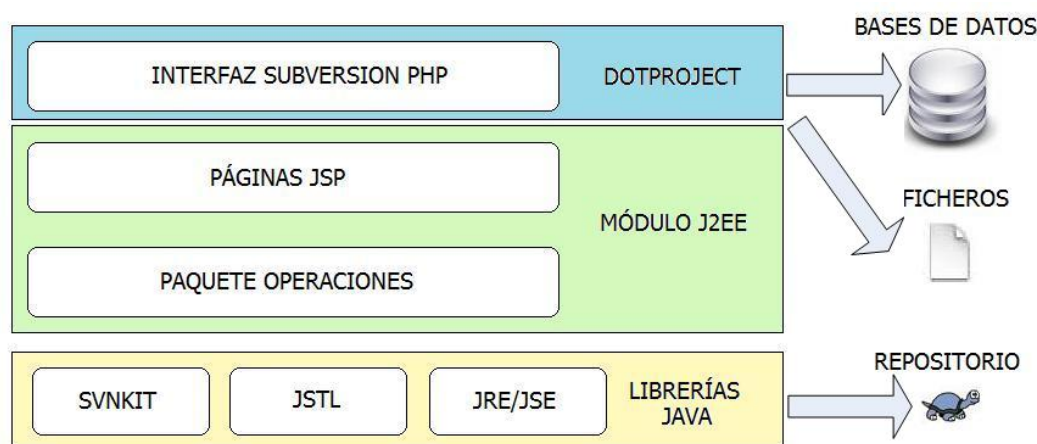


Diagrama de capas incluyendo componentes J2EE

4.5.1 Paquete de componentes J2EE

La primera parte consiste en el desarrollo de un módulo j2EE que realiza las operaciones de Subversion mediante el uso de las librerías del SVNKit.

El paquete se distribuye por separado bajo un fichero war auto contenido que incluye tanto librerías SVNKit como JSTL básicas para el correcto funcionamiento del mismo. Este módulo a su vez se constituye de la siguiente manera:

4.5.1.1. Paquete de clases java "Operaciones"

Contiene las clases java donde se realizan las llamadas a la librería SVNKit. A continuación desglosamos las diversas clases que:

- SVN.java y FSFS.java son las dos clases más importantes del paquete. Las clases implementan un manejador de funciones de SVNKit a través del cual se permiten realizar las operaciones básicas que se producen en un repositorio (Commit, Checkout, Add, etc.). La diferencia entre ambas clases es que la clase SVN está orientada a

realizar estas tareas sobre repositorios externos que se accedan mediante el protocolo "svn" mientras que las operaciones que permite realizar FSFS se acometen en el sistema local.

- *CreateRepository.java* realiza la creación de un repositorio inicial. Esta operación contrae ciertas características que no permiten su inclusión en las clases anteriores.
- *Information.java* es la clase que proporciona la información básica sobre un repositorio.
- *ArbolRepositorio.java* es una clase esencial para la visualización del repositorio en el sistema. Su propósito es generar un fichero xml con la estructura arbórea actual del repositorio. Este fichero está sujeto a una codificación especial que sigue los patrones de la librería Xloadtree (Para obtener más información sobre esta librería acudir a la sección Detalles de Implementación).
- *Mkdir.java* y *NewFile.java* son dos clases que están dispuestas para proporcionar una funcionalidad que no ha sido incorporada en última instancia. Se trata de crear archivos y carpetas sin tener que manejarlos en nuestra máquina local. Estas clases requieren de un escritor HTML (Más información buscar fckeditor).

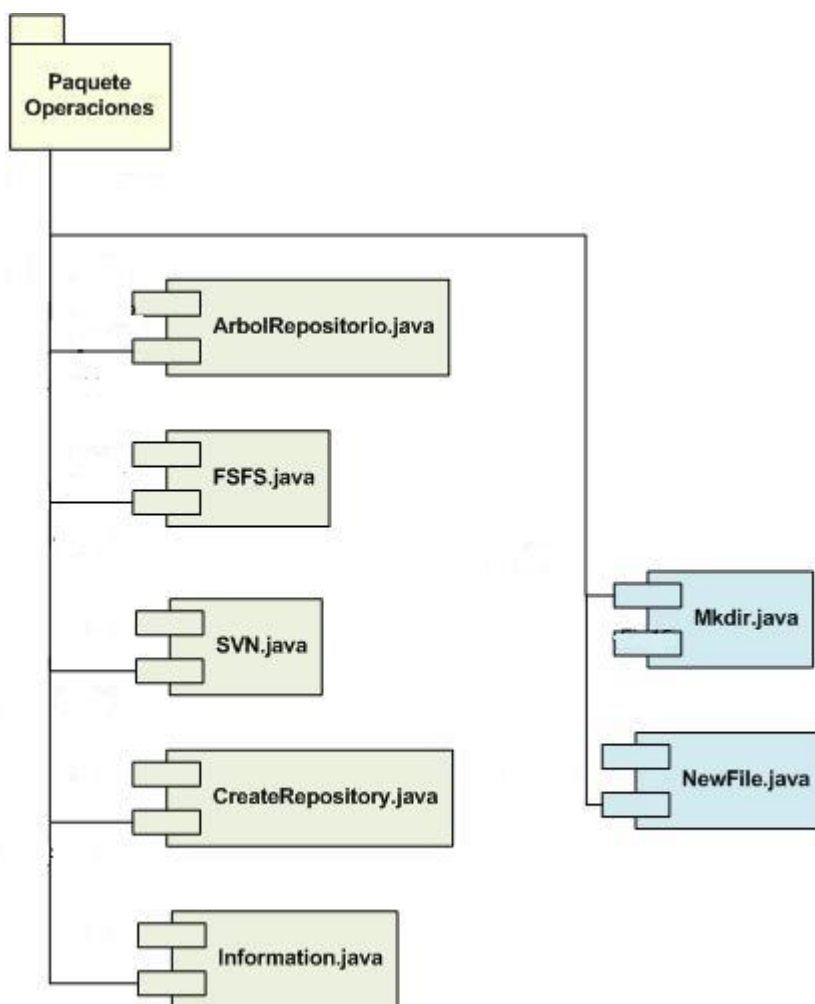


Diagrama de componentes del paquete de operaciones de subversión

4.5.1.2. Conjunto de páginas jsp

Las páginas jsp tienen por función principal realizar el enlace entre el paquete "Operaciones" y la interfaz php del sistema. Para realizar esta tarea se añaden una página por cada funcionalidad que requiere el sistema. Como ejemplo ilustrativo señalar la existencia de las páginas Commit.jsp, Checkout.jsp, etc. Para realizar el enlace la información se transmite bajo diferentes codificaciones y conversiones de formatos que permite el lenguaje HTML. Estas codificaciones son necesarias para interpretar correctamente tanto nombres de ficheros y usuarios como rutas del sistema. La mayoría de estos datos son transmitidos a su vez vía cookies o encabezados debido a que es la manera más eficiente de transmitir datos entre páginas que se manejan en diferentes puertos (servidores).

Módulo de interfaces de subversion en php

El módulo php es más anárquico en su estructura ya que se forma según la disposición de interfaces requerida por el sistema. Básicamente es un aglomerado de clases que se estructuran alrededor de una interfaz

principal. La interfaz principal es la página subversion.php, una página que muestra al usuario todas las posibles operaciones que puede realizar bajo el módulo de subversion. Esta interfaz se asocia con cada proyecto y realiza internamente las peticiones de permisos necesarias para el trabajo en el sistema local. En el esquema siguiente podemos ver la estructura planteada:

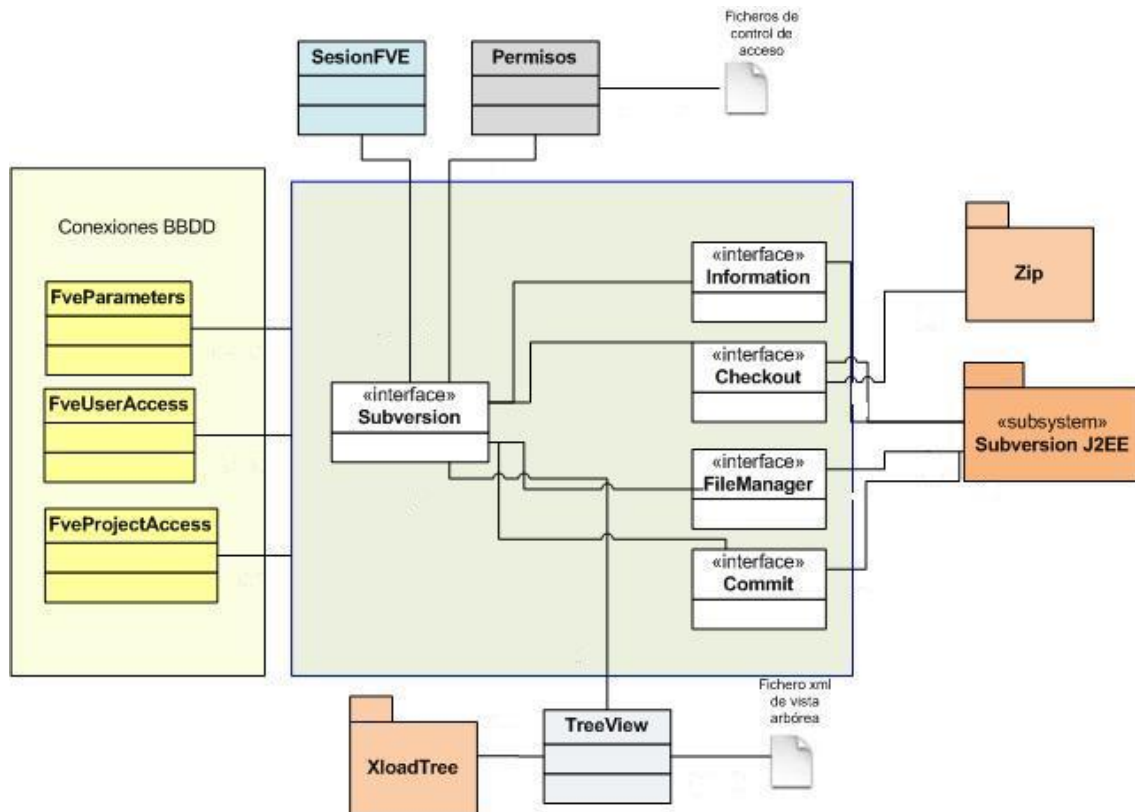


Diagrama de componentes del módulo de subversion

Sobre el esquema se pueden ver las comunicaciones con las bases de datos y la autenticación del usuario que se ligan directamente con la interfaz superior. La clase sesionFVE también se utiliza aquí para consultar los valores de sesión del usuario.

De la interfaz superior se desglosan diversas interfaces según la operación a realizar. Son estas páginas las que realizarán la comunicación con el módulo de j2EE. Por otro lado existen los módulos de apoyo que contienen las librerías Zip y XloadTree, para obtener una información más detallada de las mismas acudir a los detalles de implementación.

4.6. Detalles de implementación

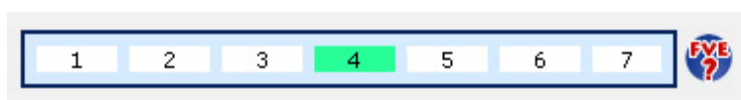
4.6.1 Wizard de instanciación de proyecto

El asistente de creación de proyectos es uno de los puntos principales de la aplicación. Permite instanciar nuevos proyectos importando su definición

desde un archivo XML creado con EPF Composer que contenga la estructura del proyecto. A través de distintos pasos, se irán configurando todos los parámetros importados para que, al finalizar el wizard, el proyecto importado tenga todos sus componentes configurados en base a las preferencias del usuario.

Este asistente consta de siete etapas, cada una con un fin determinado. Como se explicó en la sección anterior, al arrancar el asistente se crea una nueva "Sesión Wizard", que está gestionada por una instancia de la clase `sessionWizard`. Esta clase irá controlando todos los aspectos de la sesión del usuario en el asistente, como el paso en el que se encuentra o el archivo XML que ha subido.

Durante el proceso de creación del proyecto, el usuario puede *pausar* en cualquier momento la operación mediante el botón "Pause Wizard". Esto significa que el usuario podrá retomar posteriormente el asistente con la misma información con la que lo dejó. Si por cualquier circunstancia el proceso de creación se ve interrumpido, corte de luz, caída de la conexión, etc., la sesión del asistente por defecto se restaurará por el último punto que dejara el usuario activo y con el mismo estado del proyecto. Es decir, es como si hubiera pausado la sesión. Esto ocurre porque el diseño del wizard se pensó para que tuviera la mayor robustez posible y para que esto fuera realizable, el wizard confirma los cambios que el usuario realiza en cada paso tras pasar a la siguiente etapa. Es decir, cada vez que el usuario avanza o retrocede a otro paso del asistente, los cambios que hubiera realizado en el anterior paso se confirman sobre el proyecto, quedando almacenados en la base de datos. Éste método de modificación quizás tenga un mayor peso de cara a la carga de proceso, pero las ventajas que aporta a nivel de robustez hacen que compense su puesta en práctica.



Menú superior de pasos en el wizard.

Para que el sistema sepa en qué punto del asistente dejó el usuario su sesión, lo que se hace es que al crear una sesión de Wizard, se crea una fila en la tabla "fve_user_wizard_sessions" de la base de datos. Esta fila únicamente existirá durante la vida del wizard y una vez se dé por finalizado se borrará. La información que contiene esta fila es el identificador del nuevo proyecto, el usuario que lo lanzó y el paso actual en el que se encuentra o dejó el asistente. Esta fila sólo se borra al finalizar el proyecto, así que si el usuario detiene el asistente y luego lo vuelve a retomar, en el momento que se cree de nuevo la clase `SessionWizard`, ésta comprobará mirando en la tabla "fve_user_wizard_sessions" que anteriormente el usuario dejó una sesión a medias, por lo que cargará el asistente por el último punto que dejó configurado el usuario.

En cada paso del asistente los datos que se muestran de los componentes del proyecto se corresponden con la información de la base de datos. De esta forma, al movernos hacia delante y hacia atrás en el wizard, si hubiéramos realizado previamente alguna modificación en algún punto se mostrarían los cambios por pantalla, confirmando así de manera visual al usuario que las modificaciones realizadas han tenido éxito.

4.6.2 Importador XML de procesos de desarrollo de EPF Composer

Una de las tareas principales del sistema era la importación de procesos de desarrollo de EPF Composer al módulo creado en dotProject. Esta importación se realiza con la entrada únicamente de un fichero XML generado por la aplicación de origen, basado en un complejo schema y con una serie de pautas no inmediatas. Hay que destacar que el fichero es capaz de importar todas las propiedades de una librería de procesos y no solo de un proyecto específico, por lo que esta flexibilidad se traduce en una complicación de la estructura. E incluso otro factor más que complicaba el proceso de parseo del fichero fue el carácter fuertemente conexo del fichero, de modo que todas las entidades presentaban un gran número de dependencias entre sí.

Por todo ello se decidió crear una estructura modular y fácilmente extensible en el proceso de importación del fichero xml, utilizando técnicas habituales en procesamiento de lenguajes. En primer lugar se estudiaron las opciones de importación de datos xml en el núcleo de PHP, encontrando dos parsers que ayudarían a gestionar el proceso: XML Parser [PHPManual12 07] y SimpleXML [Hughes 04]. Se decidió el uso de la segunda alternativa por permitir una mayor claridad en el proceso, debido a la mayor simplificación de las primitivas disponibles y la posibilidad de acceso sencillo a funciones XPath (XML Path Language), que en nuestro caso eran necesarias para tratar los contenidos generales previamente a los procesos de desarrollo ya que el fichero XML no trae las instancias de cada entidad ordenadas.

El proceso normal de un fichero XML sigue este esquema jerárquico de funciones:

```

processContentElement($xml,$project);
└─ processArtifact($contentElement,$project);
    └─ Insert/Update:Workproduct(...);
        └─ processArtifactAttributes($contentElement,$project,$workproduct_id);
            └─ insertWorkproductAndTemplate($workproduct_id,$template_id);
└─ processRole($contentElement,$project);
    └─ Insert/Update:Role(...);
        └─ processRoleAttributes($contentElement,$project,$role_epf_id);
            └─ insertRoleResponsibleWorkproduct($role_epf_id,$workproduct_id);
└─ processTask($contentElement,$project);
    └─ Insert/Update:Task(...);
        └─ processTaskAttributes($contentElement,$project,$task_id);
            └─ insertRolePrimaryTask($role_epf_id,$task_id);
            └─ insertRoleAdditionalTask($role_epf_id,$task_id);
            └─ insertTaskAndWorkproducts($task_id,$workproduct_id,$type_dep');
└─ processTemplate($contentElement,$project);
    └─ Insert/Update:Template(...);
processMainProcess($xml,$project);
└─ processProcess($process,$project,$type);
    └─ processProcessAttributes($process,$project,$type);
includeParentAndDependenciesInTasks($project);

```

Jerarquía de funciones en importador XML.

La implementación de la importación de fichero XML presenta estas particularidades relevantes:

- Análisis en cascada de los datos parseados: Se analizan los nodos XML de la entidad más general a los contenidos más específicos, realizando un recorrido en profundidad por todos los atributos y nodos internos de un elemento antes de pasar al análisis del siguiente (separando, eso sí, los 2 módulos principales que se muestran en el esquema: Elementos generales (ContentElement) y procesos de desarrollo (Process)).
- Generación “al vuelo” de datos: El proceso de importación se realiza de manera eficiente, anticipando operaciones para poder generar en una sola lectura del resultado del analizador xml todos los datos del fichero. Esto se consigue insertando los elementos a partir del identificador que les proporciona EPF aunque el elemento aún no haya sido declarado, esperando a que esto suceda para actualizar la información del elemento, del mismo modo que se haría en un compilador. Debido a esta mejora, que únicamente necesita de una pequeña función de corrección en las dependencias entre tareas de un proceso, se consigue una mejora en la eficiencia de un 40% respecto a un procesado estándar.
- Dobles identificadores: Los elementos pueden declararse como contenidos generales del plugin, o como elementos dependientes de un proceso de desarrollo. En ocasiones interesa emplear un contenido general en un

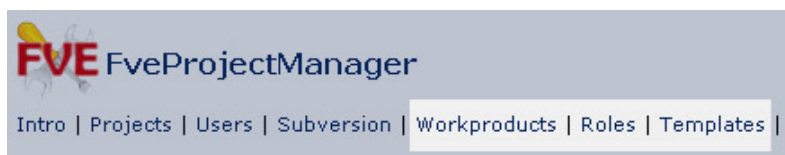
proceso de desarrollo, lo que produce una asociación entre entidades, que sin embargo continúan teniendo identificadores diferentes. Esto supone que en un análisis de una sola pasada se podrían insertar anticipadamente dos entidades que corresponden a una misma, por lo que se anota información en la base de datos sobre los identificadores genéricos y específicos tratados para que si se detecta un problema de este tipo pueda emplearse una función correctora de esta cuestión, siendo transparente de cara al usuario si la situación se ha producido o no y obteniendo el mismo resultado que si emplearan los mismo identificadores.

- Procesos con entidades como atributos: Como ya se ha indicado anteriormente, los procesos de desarrollo pueden tener relación con elementos generales (roles, tareas, etc.) o declarar sus propios elementos en el proceso. Esto implica que los atributos de los procesos pueden requerir de un recorrido más amplio que únicamente insertar los datos encontrados, y deberán llamar a funciones como `processRole` o `processTask` ya empleadas. Esto se permitirá adaptando los atributos en la llamada y reutilizando las funciones ya implementadas para procesar los elementos generales, aunque no tengan una estructura exactamente igual. Esto se consigue “relajando” ciertas restricciones del modelo por lo que se aceptarían algunas incorrecciones de acuerdo al schema XML, pero que no pueden producirse debido a que el fichero es realizado automáticamente y no escrito por un usuario y que sí se exige que tenga la estructura casi completa de un fichero xml de EPF Composer.

4.6.3 Doble funcionalidad en contenidos FVE (en núcleo de dotProject y en módulo FVE)

Algunas páginas de la aplicación son accedidas desde puntos diferentes del programa y deben de mostrar similares contenidos aunque difiriendo en algunas partes.

Más en concreto, los archivos que listan los roles (`epf/roles/list.php`), los productos de trabajo (`epf/workproducts/list.php`) y las plantillas (`epf/workproducts/list.php`) tienen dos formas de acceder a ellos. Si se les suministra un identificador de proyecto como parámetro en su llamada (`$_GET[project_id]`), la información que muestren será relativa al proyecto del que se solicita, es decir, todos los roles, productos de trabajo y plantillas del proyecto solicitado. Sin embargo, si no se especifica proyecto ninguno, se mostrará la información de todos los roles, productos de trabajo y plantillas existentes en todos los proyectos de la aplicación. Este tipo de llamadas sin parámetros se realizan en las funciones facilitadas por nuestro módulo para listar roles, productos de trabajo y plantillas.

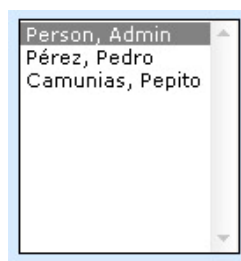


Visualización común de entidades en dotProject y fve.

Esta forma de realizar una doble llamada a un mismo archivo se utiliza con el fin de reutilizar de código de manera sencilla y evita el tener que replicar archivos cuya función difiere en apenas unas líneas.

4.6.4 JavaScript dinámicamente generado

Una de las limitaciones del paso de parámetros entre páginas de HTML con la que nos encontramos fue que no se pueden pasar entre diferentes páginas componentes 'select' con múltiples elementos como el que se muestra a continuación:



Ejemplo de usuarios en el wizard.

Por ello, la solución que optamos por realizar fue la de crear un campo oculto (tipo *hidden*) que contuviera una lista con los elementos contenidos en el select separados por comas. Por ejemplo, para el caso de la anterior imagen, se crea una lista con los identificadores de los usuarios, generando una cadena de texto con contenido similar al siguiente: "1,12,8".

Esta cadena se recibe en la siguiente página del wizard y se parsea para obtener los usuarios que debe asociar al proyecto.

Para poder realizar esta operación de construcción de una lista con los identificadores de los usuarios insertados en el select, se utiliza un método JavaScript llamado **createArray**. Este método necesita el nombre del 'select' del que se quieren obtener los elementos insertados. Con ello, modifica la lista oculta a generar insertando los elementos separados por comas.

Sin embargo, nos encontramos con otro problema y es que existen algunos puntos del wizard (para los productos de trabajo y las tareas) que hacen uso de un número indeterminado (dependiente del número de elementos que se hayan importados desde el archivo XML) de selects con opción de selección múltiple. Por lo tanto, es preciso crear un **createArray** para cada uno de ellos con el nombre que tenga el 'select' para cada elemento

importado. Como este número de elementos es variable y desconocido en un principio, no es posible escribir todos los métodos de antemano. Para solucionarlo, llevamos a cabo una técnica en la cual mediante código PHP producimos código JavaScript, ya que el código Javascript va empotrado en la propia página web. El código de esta generación dinámica de los **createArray** se muestra a continuación:

```
function createArray(form){
    <?php
        for ($j = 0; $j<$hiddenLists; $j++){
            $fieldName = $fieldNameList[$j];
            $hiddenListName = "hiddenList$j";
            echo "var insertLen = form.$fieldName.length;\n";
            echo "var i\n";
            echo "if (insertLen > 0){\n";
            echo "    form.$hiddenListName.value =
\"\"+form.$fieldName.options[0].value;\n";

                echo "        for (i=1;i<insertLen;i++){
            echo "            form.$hiddenListName.value =
form.$hiddenListName.value + \",\" +
form.$fieldName.options[i].value;\n";
            echo "        }\n";
            echo "    }\n";
        }
    ?>
}
```

Función createArray con javascript dinámico.

Como se puede apreciar, existe un 'for' que iterará tantas veces como 'selects' existan. Para cada iteración, producirá el cuerpo de un **createArray** en el que el nombre del 'select' sea el correspondiente al de la iteración.

4.6.5 Subversion

4.6.5.1 - Elección de las herramientas principales y consideraciones técnicas sobre la comunicación dispuesta entre ambas

Cuando decidimos introducir Subversion supimos rapidamente que la tarea iba a ser compleja y requerir un esfuerzo de investigación. Muchas son las herramientas que utilizamos para generar diversos prototipos y pruebas hasta por fin dar con una solución. El problema mayor provenía de que la única herramienta de las estudiadas para manipular Subversion que se ajustaba a nuestros requisitos era un framework libre basado en Java llamado SVNKit [SVNKit 07] (ya descrito en la sección 2.3). Este paquete integra diversos tipos de clases para manipular repositorios de SVN. El hecho de que las clases estuvieran en java nos guió directamente a buscar soluciones web de J2EE que pudieran trabajar con estas clases y que, a la vez, ofrecían vía web la posibilidad de interactuar con ellas. En ese

momento teníamos un conflicto, pues teníamos muy claras dos cosas para que el proyecto se llevara a cabo en los plazos requeridos: Usar dotProject como base para reutilizar toda su estructura de proyectos, usuarios, etc. que estaba escrito en php, y a la vez utilizar SVNKit a través de un servidor j2EE. Las dificultades surgían al tratar de realizar el nexo de unión entre ambas, y que no quedaran muy distantes las dos partes. Poco a poco se encontraron las opciones para realizar esto vía web, y como los dos módulos separados podrían funcionar en distintos puertos y al final serían unidos mediante simples enlaces y redirecciones transparentes al usuario. Para que todo esto fuera posible, el administrador del sistema sólo tendría que configurar el módulo php para redireccionar correctamente las páginas al módulo de j2EE. Se introdujo esta opción permitiendo así la automatización de las conexiones entre ambos servicios. Más tarde descubrimos que la comunicación entre los dos servicios era más compleja ya que no permitía cualquier tipo de información entre ellos por residir en diferentes puertos y que por tanto la mayor parte de la información que se transmitiría iba a ser enviada entre ambos sería vía cookies.

4.6.5.2 - Estructura física de los ficheros de Subversion

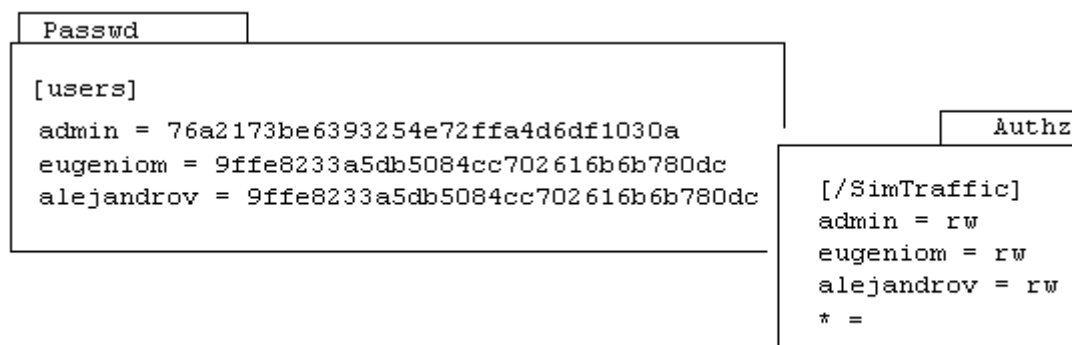
Una vez establecidas las comunicaciones y servicios surgen los problemas de cómo servir al usuario los archivos deseados del repositorio. La primera decisión fue saber de cuantos repositorios íbamos a requerir. Lo inmediato era pensar que uno por proyecto, pero el manejo de tantos repositorios se hacía impracticable y no era habitual en los casos prácticos de uso de Subversion. Descubrimos posteriormente que el uso de uno sólo con la integración de permisos basados en carpetas nos iba a proporcionar un método más asequible de integrar la estructura de proyectos bajo un mismo repositorio. Así pues, el repositorio de nuestro módulo se distribuye en carpetas con los nombres de los diversos proyectos que mantienen restringido su acceso en unos ficheros del repositorio de los cuales más adelante hablaremos.

Elegida la opción del único repositorio pensamos en trabajar directamente sobre él extrayendo información y archivos bajo demanda del usuario en el módulo php. La clase SVNKit ofrece un método para extraer uno a uno los archivos del repositorio para ser enviados, pero esta clase conllevaba un retraso inaceptable y no facilitaba el manejo del repositorio. Se optó por tener una copia de trabajo local en el sistema que se actualizase con cada cambio realizado en el repositorio. Al residir además el contenido del repositorio en forma de ficheros físicos del sistema, estos pueden ser accesibles mediante php de forma corriente sin tener que hacer peticiones al módulo de j2EE como en la anterior propuesta ideada.

Una vez definida la estructura del sistema, las interacciones entre los dos módulos, si bien fueron algo laboriosas, eran posibles e implementables. Faltaban todavía unos aspectos que resolver en cuanto a cómo se iba a transmitir toda la información del repositorio al usuario usando servicios web con los posibles problemas de subidas y bajadas múltiples de archivos, y cómo integraríamos los permisos de usuarios establecidos en dotproject para el acceso al repositorio de subversion.

4.6.5.3 - Permisos en Subversion

Para dar con la solución a los permisos recurrimos rápidamente al manual para saber bajo que archivos Subversion otorgaba permisos generales y basados en directorio. Analizamos las posibilidades, y la que más se adecuaba a nuestras peticiones era la de modificar los archivos de autenticación del servidor de svn mediante svnserve. Existía otra opción mediante archivos de configuración de autenticación si usábamos un apache como servidor de Subversion, pero esto ocasionaba problemas debido a la encriptación de los ficheros de apache y la obligación al usuario de instalar apache en el sistema. En cambio la opción de manejar los archivos del svnserve era mucho más factible. La implementación era casi directa, por cada proyecto había que generar los usuarios del sistema en la configuración de contraseñas de Subversion, y establecer una relación de usuarios permitidos en cada proyecto. Obteníamos un esquema como el que sigue:



Esquema de archivos Passwd y Authz.

En el fichero passwd escribíamos la relación de contraseñas con los usuarios(aquí vienen encriptadas), mientras que en el fichero de Authz se escriben las relaciones de permisos de usuarios con su proyecto (en la imagen * = significa que el resto de usuarios no tiene ningún permiso ni de escritura ni de lectura).

En el módulo de php todos estos archivos serían comprobados para realizar ciertas acciones sobre la copia de trabajo local y sus ficheros, ya que estos no tienen ninguna restricción de lectura aparente al igual que para cambios

realizados mediante el módulo de j2EE. En cuanto a los cambios realizados en el repositorio via el servidor de svn (svnserve), es éste quien pregunta por los permisos al usuario que trata de acceder via consola.

4.6.5.4 - Transmisión de los ficheros de Subversion vía web

Es bien sabido que los formularios html no ofrecen más que la posibilidad de subir un archivo por cada campo de fichero introducido. De manera similar tampoco se puede descargar más de un fichero a la vez. Las diferentes soluciones que surgieron se agrupaban en dos:

- La introducción de applets certificados en el sistema de dotproject, opción que descartamos por el impacto visual, de permisos y otros aspectos que perjudicaban el sistema como las alertas de seguridad que plantearía la aplicación.
- Aprovecharse de los sistemas de compresión y descompresión de archivos que disponemos en php para realizar mediante la descarga o subida de un archivo comprimido obtener todo el contenido de un proyecto, carpeta, etc. Se implementó esta opción con la ayuda de una librería de licencia GNU-GPL llamada PclZip. La inclusión de esta librería vino por el hecho de que php no contiene en ninguna de sus distribuciones funciones de compresión (si de descompresión) a nivel de carpetas y ficheros (Php sí que ofrece en cambio compresión de cadenas y otros elementos que son las que utiliza esta librería para realizar la tarea).

4.6.5.5 - Visualización del repositorio con la ayuda de árboles XloadTree

Estudiando la librería SVNKit, descubrimos que un método sencillo de listar los contenidos de un repositorio y de escribirlos con la estructura jerárquica de exploración deseada bajo un archivo xml. Aquí es donde aparece el uso de la librería XloadTree. Se trata de una pequeña librería escrita en javascript y de licencia GNU GPL asociada a Apache, que permite representar estructuras arbóreas en páginas html. Sin entrar mucho en detalle, un posible ejemplo de xml:

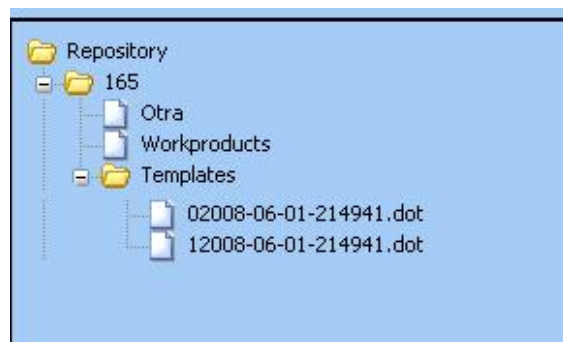
```

<?xml version="1.0" ?>
- <tree>
- <tree text="165" action="../../index.php?m=fve&pag=SubversionFileManager&path=165" target="_top">
  <tree text="Otra" action="../../index.php?m=fve&pag=SubversionFileManager&path=165/Otra" target="_top" />
  <tree text="Workproducts" action="../../index.php?m=fve&pag=SubversionFileManager&path=165/Workproducts"
    target="_top" />
- <tree text="Templates" action="../../index.php?m=fve&pag=SubversionFileManager&path=165/Templates"
  target="_top">
    <tree text="12008-06-01-214941.dot" action="./Temp/TempWC/165/Templates/12008-06-01-214941.dot"
      target="_top" />
    <tree text="02008-06-01-214941.dot" action="./Temp/TempWC/165/Templates/02008-06-01-214941.dot"
      target="_top" />
  </tree>
</tree>
</tree>

```

Código de ejemplo de fichero a leer por XTreeMap.

Y su representación asociada sería:



Representación del árbol de un repositorio de ejemplo.

Además cada objeto de este árbol puede contener un vínculo, que es básicamente como definir enlaces en html. Esto nos permitió una gran flexibilidad para mostrar el repositorio de subversion y visualizar los archivos.

5. Organización del proyecto

5.1 Planificación de proyecto

El trabajo del proyecto se estructuró en 3 fases, cada una con iteraciones en las que se fueron cumpliendo objetivos del proyecto de manera gradual y refinando los productos generados en las tareas de iteraciones anteriores. Se deja constancia es este apartado de las entregas realizadas y los contenidos tratados en cada fase e iteración.

5.1.1 Elaboración

5.1.1.1 - Iteración 0: Definición e investigación del proyecto

(15 Octubre – 15 Noviembre)

En esta primera iteración del proyecto se realizó un estudio de las primeras herramientas propuestas en la propuesta del proyecto y se comenzó el trabajo de obtención de requisitos de la aplicación. Desde el primer momento además se buscó adentrarse en las herramientas y en esta primera iteración ya se genera un prototipo. Esta iteración terminó con la entrega de la primera versión de los siguientes documentos:

- Software Requirement Specification (SRS).
- Propuesta de casos de uso del proyecto.
- Estudio de Workflows
- Estudio de Gestores de proyectos.
- Estudio de sistemas de control de versiones.
- Estudio de EPF Composer.
- Documento de investigación y prototipado (incluyendo el desarrollo del primer prototipo de gestión de tareas y dependencias).

5.1.1.2 - Iteración 1: Análisis detallado de herramientas implicadas

(15 Noviembre – 20 Diciembre)

La siguiente iteración tuvo como objetivo restringir el área de búsqueda y realizar un estudio mucho más profundo de las herramientas que parecieron más adecuadas (aunque en este momento aún no decididas) de cara a ser integradas en el sistema: EPF Composer, dotProject y Subversion. Esta iteración terminó con una decisión más firme sobre las

herramientas que estarían implicadas en el proyecto, con un nuevo prototipo, y con la siguiente documentación.

- Instalación de un repositorio Subversion y configuración de herramienta Trac para visualización del código.
- Informe de EPF Composer: Exportación de datos y conexión con MsProject y dotProject.
- Documento sobre el prototipo de interfaz de usuario (incluyendo el desarrollo del segundo prototipo).

5.1.1.3 - Iteración 2: Arquitectura y diseño del sistema

(8 Enero –1 Febrero)

En esta iteración se completaron los trabajos de las anteriores iteraciones, finalizando la fase de elaboración del proyecto. Se realizó un mayor énfasis en las funcionalidades que estaban menos claras, como eran los usos y las herramientas que ayudarían a gestionar Subversion, y se preparó la fase de construcción desde diferentes puntos de vista: Se instaló una wiki para favorecer el trabajo colaborativo en un servidor propio, se habilitó un espacio en SourceForge para alojar el código fuente del proyecto y se realizaron diagramas y documentos que insistían de forma más clara en la forma de llevar a cabo los requisitos ya especificados y prácticamente estables. Además, se siguieron completando cuestiones ya tratadas, como el análisis de la arquitectura de dotProject, y se generaron los primeros esquemas entidad relación de la base de datos. Por último, debido a la elección de PHP como lenguaje principal de implementación se intensificó la fase de formación en este lenguaje debido a la falta de experiencia de los componentes en PHP.

En esta iteración se generaron y entregaron los siguientes documentos:

- Elaboración de esquemas y diagramas de los interfaces del módulo.
- Funcionamiento de Subversion bajo PHP o JBoss.

5.1.2 Construcción

5.1.2.1 – Iteración 3: Creación y modificación de entidades principales y funciones Subversion

(1 Febrero – 3 de Marzo)

Al comienzo de esta iteración se completó el primer diseño de la base de datos, teniendo por tanto una estructura, experiencia en prototipos, y documentación suficiente para comenzar con los trabajos de implementación. Los primeros pasos fueron encaminados a definir la sesión

de la aplicación y a permitir la inserción y modificación de entidades generales en la base de datos a través de un interfaz web en dotProject. En esta fase se permitió la creación de roles, tareas y productos de trabajo aunque sin establecer todas las relaciones definidas en EPF Composer. Se realizaron además esfuerzos en la gestión de usuarios y los permisos que se asignarían a cada rol definiendo las primeras funcionalidades básicas.

Además se realizaron trabajos sobre el módulo de Subversion implementando las funciones habituales de trabajo con un repositorio, pero en paralelo con el resto de la aplicación y dejando pendiente la integración de las dos partes.

En esta iteración se generó como documentación:

- Diseño de base de datos, listando todas las tablas necesarias con su script y función.
- Convenios de dotProject a seguir.
- Actualización de información sobre la arquitectura de dotProject.

5.1.2.2 – Iteración 4: Importación de procesos de desarrollo y asistente de instanciación de proyecto

(3 Marzo – 31 Marzo)

En la segunda iteración de esta fase se produjeron numerosos avances en la construcción, desarrollando el primer importador de procesos de desarrollo a partir de un fichero XML (aunque solo se permitía importar un proceso de desarrollo). Además, se complementó la importación con la creación de un Wizard de instanciación de proyectos que a partir de esta información importada, insertada en la base de datos, permitía la configuración del proyecto asignando los roles a usuarios y éstos a los productos de trabajo y tareas.

Se generó como documentación:

- Actualización de la base de datos de la aplicación, con algunos cambios en el diseño inicial.
- Primera versión de manual de usuario con las funcionalidades existentes.

5.1.2.3 – Iteración 5: Gestión completa de entidades e integración de Subversion

(31 Marzo – 30 de Abril)

En esta tercera iteración de la fase de construcción se realizaron numerosos cambios orientados a permitir la creación y actualización de

todas las entidades y sus relaciones. Para ello se realizaron módulos de edición de cada entidad basados en los trabajos realizados en el wizard de creación de proyectos, y comenzaron a integrarse ciertas funcionalidades e información del módulo FVE en el módulo de tareas y proyectos de dotProject, para hacer más visibles los cambios e inserciones realizadas sobre la base de datos. Se continuaron generando consultas útiles y frecuentes para completar las clases de acceso a las tablas de la base de datos descritas en el apartado de desarrollo, provocando una mayor modularidad del código y una mejor ordenación de los datos para el usuario. Además y por otra parte, se solucionaron numerosos problemas en el apartado de Subversion, permitiendo que las operaciones funcionaran aunque no se estuvieran ejecutando en un entorno local.

Además, los problemas en el wizard e importador de proyectos se solucionaron, favoreciendo la robustez del sistema e implementando aspectos como transaccionalidad de la base de datos o mejoras en la información de sesión.

Esta iteración supuso la creación y actualización de los siguientes documentos:

- Actualización del documento de base de datos.
- Documento de transacciones de base de datos.
- Documento de estado del módulo de Subversion.

5.1.3 Depuración y pruebas

5.1.3.1 – Iteración 6: Reorganización de contenidos, funcionalidades nuevas relativas a lo planificado y solución de errores

(30 Abril – 27 de Mayo)

A partir de esta iteración los esfuerzos en depuración y pruebas se incrementaron notoriamente, restando parte de esfuerzo a la creación de nuevas funcionalidades (de forma más clara en la segunda parte de la iteración). En esta iteración se terminaron los asistentes que permitían la creación y actualización de roles, tareas, productos de trabajo y plantillas con todas las relaciones requeridas por la aplicación (incluyendo subida de plantillas a la aplicación, utilizando subida de ficheros en ftp y en repositorio Subversion). Además sufrieron una gran transformación las páginas que mostraban el estado de una instancia de rol, workproduct, plantilla, tarea y proyecto, reorganizando la información y generando un interfaz más agradable. Se trabajó también en esta iteración en el trabajo diario con la aplicación, permitiendo un mayor seguimiento de las actividades

permitiendo comitear ficheros al repositorio Subversion, actualizar el estado de un producto de trabajo si el usuario tenía permisos necesarios, etc.

Se trataron de solucionar todos los problemas existentes y mencionados en las fases de pruebas, además de realizar un trabajo de limpieza de la aplicación, eliminando información de depuración, adaptando títulos o descripciones incompletas y traduciendo contenidos que aún no estuvieran en inglés.

En esta iteración se realizó un despliegue completa de la aplicación en un servidor de la Facultad de Informática proporcionado para permitir las pruebas de manera más sencilla, y con esto se realizó una actualización del manual de instalación y de usuario.

En esta fase se actualizaron prácticamente todos los documentos implicados en la memoria, ya que se comenzó a recopilar información de anteriores iteraciones y actualizar la información necesaria, principalmente elaborando información para el apartado de Desarrollo.

5.1.3.2 – Iteración 7: Solución de errores, revisión de documentación y generación de ejemplos

(27 Mayo – 16 de Junio)

En esta iteración se hizo de nuevo un esfuerzo en corregir las funcionalidades que no se comportaban completamente bien. El módulo que más cambios sufrió fue el de Subversion, corrigiendo el comportamiento de la aplicación ante proyectos grandes, facilitando las operaciones de commit y subida de ficheros al repositorio. Además fueron revisadas las funcionalidades realizadas respecto a los informes de errores y requisitos que teníamos para comprobar que los objetivos estaban cubierto y solucionar errores.

De cara a generar el producto final fueron generados casos de ejemplo en la aplicación, principalmente creando la metodología de desarrollo aplicable a proyectos de Ingeniería del Software e importando el proyecto y subiendo documentos respectivos a la aplicación. También se incluyeron ejemplos sobre otras metodologías existentes, probando el uso de las funcionalidades implementadas y que el resto de funcionalidades eran capaces de trabajar con los datos nuevos (diagramas de Gantt, reports, etc.). Con todo esto fue completado el manual de usuario e instalación y generada una versión disponible para ser descargada de SourceForge.

Con respecto a la elaboración de la memoria del proyecto, esta iteración fue clave en estructurar y uniformizar los contenidos existentes, además de generar los apartados que necesitaban de nueva información. El

resultado final, por tanto, de esta iteración, fue el presente documento y la versión completa del programa.

5.2 Reparto del trabajo

El proyecto ha sido realizado por los tres integrantes del grupo que firman esta memoria, realizando trabajos comunes en la mayoría de iteraciones aunque en ocasiones especializándose cada uno de los integrantes y actuando como responsables de tareas específicas. Aprovechando el esquema de iteraciones anterior se va a realizar una descripción general de los trabajos de cada componente del grupo por orden alfabético.

Ignacio Estirado Casado ha sido el responsable de los estudios sobre la plataforma EPF Composer, estando sus trabajos orientados a la generación de procesos de desarrollo en la herramienta, empleando procesos existentes, y posibilitando la importación de los procesos de desarrollo a EPF Composer a través del fichero XML, dirigiendo la documentación respectiva. Además de esto, ha realizado una participación activa en los documentos relativos a la elección de un gestor de proyectos, propuesta de casos de uso, primeros estudios de sistemas de control de versiones, convenios a seguir en dotProject, diseño de la base de datos y manual de usuario.

Íñigo Fábregas Martínez ha sido el responsable de la integración del sistema de control de versiones Subversion en la aplicación, estando sus trabajos orientados a la investigación de las opciones disponibles, la configuración del entorno necesaria y estableciendo la comunicación entre los contenidos realizados en J2EE y PHP, dirigiendo la documentación respectiva. Además de esto, ha realizado una participación activa en los documentos relativos al estudio de Workflows, estudio de metodologías en EPF Composer, casos de uso, actualización del diseño de la base de datos y manual de usuario.

Alejandro Vilela Palacios ha sido el responsable de la integración en un gestor de proyectos de una herramienta capaz de gestionar las nuevas funcionalidades, realizando trabajos orientados al estudio de los gestores de proyectos Open Source disponibles, prototipado en la aplicación dotProject y estudio de la arquitectura de dotProject, dirigiendo la documentación respectiva. Además de esto, ha realizado una participación activa en los documentos relativos a la definición del proyecto, como SRS y requisitos funcionales, casos de uso, actualización del diseño de la base de datos, estudio de transaccionalidad en la base de datos, manual de instalación y manual de usuario.

En las tareas de implementación de la aplicación principal en dotProject se han dividido las funcionalidades de forma menos clara entre

los componentes del grupo, estando generalmente más de un usuario dedicado en cada tarea aprovechando el sistema de control de versiones utilizado. Así, se han realizado trabajos equivalentes repartiendo la gestión de entidades de EPF Composer en dotProject, realizando las clases de acceso a la base de datos, elaborando los listados y visión de elementos y corrigiendo fallos mediante planificaciones más frecuentes y diversificadas, más orientadas al trabajo en periodos de 2-4 días. En general, además, se han dividido las tareas asociadas a aspectos diferentes a éstos, como los despliegues de la aplicación, elaboración de documentos de descripción de cada entrega, instalación de herramientas de uso interno como sistema wiki o registro en webs necesarias como SourceForge, listas de correo y foros de herramientas implicadas, etc.

Además la memoria está escrita por los tres componentes del grupo, desarrollando un trabajo individual más enfocado a los trabajos desempeñados ya descritos pero revisando, actualizando y completando secciones de todos los apartados realizados por todos los componentes.

6. Ejemplo de uso

6.1 Introducción

Se han desarrollado de cara a la depuración y prueba del proyecto, como es lógico, un gran número de ejemplos. No obstante, de cara a la entrega final, se han generado instanciaciones de proyectos utilizando los tres procesos de desarrollo por defecto de EPF Composer y un ejemplo completo de la funcionalidad del sistema que es el que se va a mostrar en este apartado.

El ejemplo completo del sistema consiste en mostrar cómo se habría podido utilizar esta herramienta para la realización del proyecto SimTraffic realizado por los 3 componentes del grupo en la asignatura Ingeniería del Software en el curso 2006/2007. Para ello, se ha instanciado un proyecto importando como proceso de desarrollo una metodología nueva acorde a la planificación de los proyectos de Ingeniería del Software (consultar para más detalle en la sección 2.2.6), se han introducido los documentos que se generaron en esa asignatura y se ha hecho una distribución de tareas, fechas y usuarios similar a la realizada en la asignatura (en su momento, una planificación manual, con los contenidos en una wiki en su mayoría y con dificultades de gestión de fechas y división de trabajo).

El ejemplo muestra de forma clara en este documento los hitos conseguidos y una parte de la planificación realizada y los resultados de la aplicación en la gestión de un proyecto concreto y existente.

6.2 Instanciación de proyecto

En primer lugar se rellenó la información general del proyecto, en nuestro caso de ejemplo SimTraffic, con una duración de un curso académico (de octubre de 2006 a junio de 2007), definiendo un administrador para el proyecto e introduciendo una descripción general con la que se asociará el proyecto:

Step 1: Project creation.

In this phase, you have to introduce the necessary data to instance a new project into the application. Please, fill the next form in order to proceed to the next step. If you need to introduce further information, please, do it later at "Project Tab".

Project name:

Company:

Owner:

Start date:

Finish date:

Description:

Simulador de tráfico vehicular realizado para la asignatura Ingeniería del software por un grupo de 20 personas.

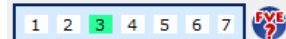
Ejemplo de uso: 1. Instanciación paso 1.

Previamente se habían dado de alta a todos los usuarios que iban a participar en este proyecto utilizando las funciones de creación de usuarios. Así, una vez hecho esto, se puede cargar el fichero XML que contiene nuestra metodología y nos saldrían en el siguiente paso todos los usuarios disponibles, pudiendo agregar a los 20 integrantes del grupo como trabajadores del proyecto en nuestro módulo.

Project Creation Wizard

The following wizard will allow you to create new projects giving you the necessary help in order to configure your projects according to your needs.

Wizard session previously created... Loaded. The actual step is 2



Opening XML file.....
Done.

Processing Libraries.....

Method Library found: 0

Method Library found: _mHWBoCgtEd26b8uzR3Azqg

Method Library found: SimTrafficProcess

Done.

Processing Plugins.....

Import plugin: SimTrafficProcess with description: Main plugin that defines orders and authors Ignacio Estirado, Íñigo Fábregas, Alejandro Vilela, with the experience of other 17 workers in SimTraffic project..

Done.

Processing Content Elements..... Done.

Processing Processes.....

Parser ends.

Done.

Finished .

Step 3: Configuring users.

Once you have created your project, you have to configure the users that will participate in it. Use the following form to select the users from the system that you want to work on the new project.

Available users:

Users that will work in the project:

Ejemplo de uso: 2. Instanciación paso 3.

En el siguiente paso definimos los roles que puede asumir cada usuario. Estos datos son difíciles de saber a priori, por lo que en un proyecto real podrían incluirse sólo los que estuvieran más claros y modificarlos más adelante. En cualquier caso, como en este caso la planificación se hace a posteriori se asignan todos los roles que en algún momento desempeñó un usuario. En este paso ya nos damos cuenta de que debido a la falta de uso de una herramienta de organización como ésta ciertos repartos y asignación de roles realizados en la asignatura tuvieron algunos desequilibrios. Se muestran en la imagen la asignación para 2 roles de ejemplo.

The screenshot displays two panels from the FVE Project Manager interface, illustrating the role assignment process.

Panel 10: Role: NonDomainInvestigator

- Description:** This role should investigate anything that is not relationated with the domain, as languages to implement the software, special characteristics of its, editors, etc. This involves in a Traffic simulator which is the better language to implement it, how to implements multithreaded applications, etc..
- Available users:** A dropdown menu showing a list of users: V., Alejandro, E., Ignacio, C., Alfonso, G., David, O., Luis, B., javier, F., Iñigo, and M., Eugenio. The user "V., Alejandro" is currently selected.
- Assigned users:** A list of users currently assigned to this role: E., Ignacio, C., Alfonso, G., David, O., Luis, B., javier, F., Iñigo, and M., Eugenio.
- Buttons:** "Add user to role" and "Remove user from role".

Panel 11: Role: Tester

- Description:** His role should implement automatic tests in the project to make easy depuration and refactorization..
- Available users:** A dropdown menu showing a list of users: M., Eugenio, M., Nacho, G., Juan, and O., Luis. The user "M., Eugenio" is currently selected.
- Assigned users:** A list of users currently assigned to this role (currently empty).
- Buttons:** "Add user to role" and "Remove user from role".

Ejemplo de uso: 3. Instanciación paso 4.

En el siguiente paso se decidieron los responsables de los workproducts importados, pudiendo seleccionar únicamente los usuarios de los roles especificados como responsables.

4 Workproduct: **Estimation**

Configuring Users
Responsible: **F., Pablo** (Leader of Subgroup)

Modifiers

Templates
Do you want to upload and associate a template file for this workproduct?: ☐ Yes: Name: Description:

5 Workproduct: **User Manual**

Configuring Users
Responsible: **V., Alejandro** (Documentor)

Modifiers

Templates
Do you want to upload and associate a template file for this workproduct?: ☐ Yes: Name: Description:

6 Workproduct: **Formal Technique Revisions**

Configuring Users
Responsible: **O., Luis** (Reunion Secretary)

Modifiers

Templates
Do you want to upload and associate a template file for this workproduct?: ☐ Yes: Name: Description:

Ejemplo de uso: 4. Instanciación paso 5.

Y finalmente, el paso que más tiempo lleva en el wizard, siendo una tarea que frecuentemente no se realizará en un solo momento, se realiza la asignación de usuarios a tareas, de acuerdo a los roles establecidos en EPF Composer como usuario primario y adicional, completando también las fechas de comienzo y fin de cada tarea. Se muestra un ejemplo de tarea importada y configurada.

9 Task: **Update case uses**
Description:

Configuring Users
Primary: **F., Íñigo** (Documentor)

Additional
Role: *Leader of Subgroup*
Available users: **F., Pablo** Add user
Modifier users: **E., Ignacio**
C., Alfonso Remove user
Role: *InspectorDocumentation*
Available users: **V., Alejandro** Add user
Modifier users: **R., Raquel** Remove user

Adjusting Dates
Start date: **17/12/2006** 11 : 00
Finish date: **22/12/2006** 12 : 00
Duration: **34** hours Calculate Duration

Linked Workproducts

Name	Description	Dependency Type
UseCases Document	A use case is a description of a system's behaviour as it responds to a request that originates from outside of that system. The use case technique is used in software and systems engineering to capture the functional requirements of a system.	mandatoryInput
UseCases Document	A use case is a description of a system's behaviour as it responds to a request that originates from outside of that system. The use case technique is used in software and systems engineering to capture the functional requirements of a system.	output

Ejemplo de uso: 5. Instanciación paso 6.

Con estos pasos se terminaría el wizard de creación de proyectos.

6.3 Visión y trabajo en el proyecto

Esta aplicación habría resultado muy útil para favorecer el conocimiento general del estado del proyecto y las tareas asignadas. Debido a la inexperiencia en la gestión de proyectos al cursar esa asignatura, y al número alto de trabajadores implicados, hubo cuestiones de organización mejorables que habrían significado una mejora del producto final en todas las fases del proyecto y haber evitado cargas de trabajo desproporcionadas en momentos concretos del proyecto.

En primer lugar se mostrará la información general del proyecto a la que todos los trabajadores tendrían acceso, aprovechando las nuevas funcionalidades y la compatibilidad del módulo con la funcionalidad de dotProject. Se muestra una visión general del proyecto.

EPF Info	
Edit EPF Project info	
Subversion repository path:	SimTraffic
Guideline link (EPF):	http://fvesolutions.es/epf/SimTr001/
Plugin used (EPF):	SimTrafficProcess
Plugin description (EPF):	Main plugin that defines orders
Plugin authors:	Ignacio Estirado, Íñigo Fábregas, Alejandro Vilela, with the experience of other 17 workers in SimTraffic project.
Users that could work in this project	
User 1	eugeniom
User 2	alfonsoc
User 3	alejandrov
User 4	inigof
User 5	ignacioe
User 6	nachom
User 7	nachoc
User 8	karena
User 9	raquelr
User 10	davida
User 11	davidg
User 12	juang
User 13	ulrichh
User 14	carlost
User 15	ivanr
User 16	pablof
User 17	sergiop
User 18	javierb
User 19	luiso
User 20	antoniog

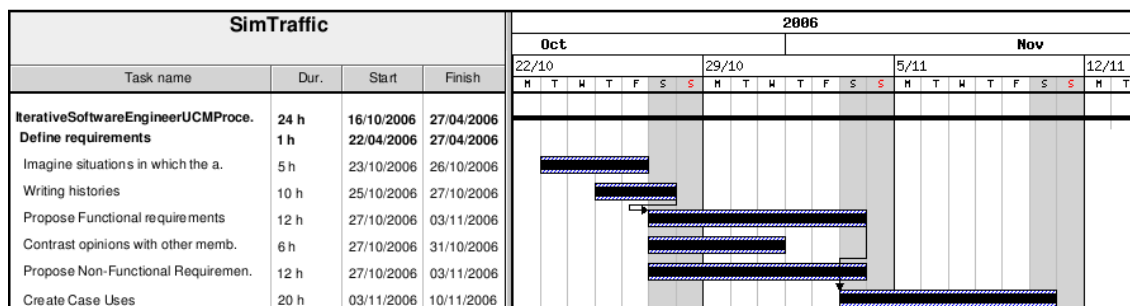
Ejemplo de uso: 6. Ver proyecto.

Para el proyecto podría consultarse la planificación de las tareas importadas (que mantiene la jerarquía y dependencias definidas en EPFComposer). Esto eliminaría las dudas que surgieron en el proyecto durante su desarrollo sobre las fechas de entrega o las estimaciones realizadas, ya que en un grupo tan grande de desarrolladores como en esa asignatura es complicado que todos tengan clara la planificación del proyecto, y haber podido llevar un mejor registro de las horas empleadas en cada tarea.

100%	Propose an idea for a project	admin	-	16/10/2006 10:00 am	16 hours	20/10/2006 02:00 pm
100%	Make a brainstorming	admin	alejandrov (100%) (+19)	16/10/2006 10:00 am	3 hours	17/10/2006 11:00 am
100%	Get support for ideas	admin	pablof (100%)	17/10/2006 10:00 am	6 hours	19/10/2006 11:00 am
100%	Discuss the more interesting ideas	admin	ignacioe (100%)	17/10/2006 10:00 am	6 hours	19/10/2006 11:00 am
100%	Vote ideas	admin	alejandrov (100%)	20/10/2006 01:00 pm	1 hours	20/10/2006 02:00 pm
100%	Define requirements	admin	(100%)	23/10/2006 10:00 am	65 hours	10/11/2006 10:00 am
100%	Imagine situations in which the application could be useful	admin	nachoc (100%)	23/10/2006 10:00 am	5 hours	26/10/2006 11:00 am
100%	Writing histories	admin	davidg (100%)	25/10/2006 09:00 am	10 hours	27/10/2006 12:00 pm
100%	Contrast opinions with other members	admin	ignacioe (100%)	27/10/2006 10:00 am	6 hours	31/10/2006 11:00 am
100%	Propose Functional requirements	admin	eugeniom (100%)	27/10/2006 10:00 am	12 hours	03/11/2006 12:00 pm
100%	Propose Non-Functional Requirements	admin	davidg (100%)	27/10/2006 10:00 am	12 hours	03/11/2006 12:00 pm
100%	Create Case Uses	admin	inigof (100%)	03/11/2006 10:00 am	20 hours	10/11/2006 10:00 am

Ejemplo de uso: 7. Lista de tareas.

El módulo de dotProject que permite visualizar diagramas de Gantt del proyecto requeridos en la asignatura es compatible con las tareas creadas por FVE, por lo que introducidos los datos en esta herramienta no se habría necesitado trabajar con MsProject de forma paralela (evitando replicar información con el coste asociado en esfuerzo y las posibles inconsistencias en la planificación). Se muestra para una iteración el diagrama de Gantt resultante:



Ejemplo de uso: 8. Diagrama de gantt de una iteración.

También se muestra la visión general de una tarea concreta, una información que aumenta de forma considerable la organización de las tareas y determina qué documentos deben generar los usuarios asignados a la tarea de manera inequívoca. Esta es la información que proporciona por defecto dotProject (que al introducir los nuevos datos del módulo en tablas de dotProject siguen mostrando toda la información pues se mantiene la compatibilidad):

Details		Assigned Users	
Project:	SimTraffic	Eugenio M.	eug@m.es
Task:	Propose Functional requirements	Dependencies	
Task Parent:	Define requirements	Writing histories	
Owner:	admin	Tasks depending on this Task	
Priority:	normal	Create Case Uses	
Web Address:		Improve and Redefine Case Uses	
Milestone:	No	Description	
Progress:	100%		
Time Worked:	0		
Dates and Targets			
Start Date:	27/10/2006 10:00 am		
Finish Date:	03/11/2006 12:00 pm		
Expected Duration:	12 hours		
Target Budget \$:	0.00		
Task Type :	Unknown		

Ejemplo de uso: 9. View Task (dotProject).

Y esta es la información que proporciona el módulo sobre la tarea:

EPF Info

Edit EPF Task info

EPF task Info

Task Type

task

Additional Users

alejandrov	Project Manager
karena	Reunion Secretary
carlost	DomainInvestigator
javierb	DomainInvestigator
davidg	DomainInvestigator
nachoc	DomainInvestigator

Responsible User

eugeniom Analyst

Associated Workproducts

Name	TypeDependency	Status	Revised by
Stories	mandatoryInput	80 %	carlost
Software Requirements Specification	mandatoryInput	100 %	antonio
Software Requirements Specification	output	100 %	antonio
RiskList	optionalInput	30 %	javierb
RiskList	output	30 %	javierb
Functional Requirements	output	90 %	alfonsoc

Workproduct "Status" Key:
= 0 - 20%
= 20 - 40%
= 40 - 60%
= 60 - 80%
= 80 - 100%
= Finished

Workproduct "Revised by" Key:
= None
= Any
= Modifier user
= Responsible user

Ejemplo de uso: 10. View EPF Task.

Existen por supuesto listados de todos los contenidos importados, sirviendo de referencia para los usuarios. Así, pueden consultarse listados de los roles de un proyecto, con nombre, descripción y los usuarios que pueden desempeñarlo, mostrando un fragmento en la siguiente imagen:

	InspectorDocumentation	His function is supervise the documentation that others generated. It's easy lose perspective of your goal if you work alone, so this inspectors treat to avoid this and warns you when something is wrong.	raquelr alejandrov juang nachoc sergiop carlost
	InspectorCode	His function is supervise the source code that others generated. It's easy lose perspective of your goal if you work alone, so this inspectors treat to avoid this and warns you when something is wrong.	nachom davidg inigof ignacioe eugeniom
	DomainInvestigator	This role should investigate other tools, specific characteristics, restrictions, singularities, etc. of the domain of the application. This involves in a Traffic simulator conditions of traffic, formats of representing maps, study of traffic lights behaviour, other traffic simulators that exist previously, etc.	carlost javierb davidg nachoc alfonsoc luiso
	NonDomainInvestigator	This role should investigate anything that is not relationated with the domain, as languages to implement the software, special characteristics of its, editors, etc. This involves in a Traffic simulator which is the better language to implement it, how to implements multithreaded applications, etc.	alfonsoc ignacioe davidg luiso javierb inigof eugeniom
	Tester	His role should implement automatic tests in the project to make easy depuration and refactorization.	nachom juang luiso

Ejemplo de uso: 11. Listado de Roles.

Además, cada usuario habría sabido a quien referirse cuando tuviera alguna duda sobre un producto de trabajo o tarea, consultando el usuario responsable o primario respectivamente y los posibles modificadores o adicionales. Esto habría supuesto un menor sobreesfuerzo del Project Manager durante el proyecto, pudiendo encargarse más de gestionar el proyecto y menos de mediador entre trabajadores. Se muestra una pantalla

que muestra un fragmento de la página que muestra los detalles de un producto de trabajo, en este caso del documento de riesgos, en el que se observan el usuario responsable y modificadores en SimTraffic de un producto de trabajo, la plantilla asociada y la relación con tareas y roles.

Show information about workproduct RiskList

This is specific information about this workproduct and his relations with tasks, roles and users.

General information:

Id: 501




Name: RiskList

Description: This artifact is a list of known and open risks to the project, sorted in order of importance and associated with specific mitigation or contingency actions.

Part of project: SimTraffic

[Edit general information](#)

Template associated:

Template name: RiskListIS Risk categorized by type and priority.   

[Edit associated template information](#)

Role's relations:

User responsible: nachom

Role responsible: QAInspector

User modifier 1 : javierb

Role modifier 1 : Documentor

User modifier 2 : sergiop

Role modifier 2 : Documentor

[Edit roles&tasks information](#)

[Edit users responsible and modifiers](#)



Task's relations:

Task	Relation
Propose Functional requirements	optionalInput
Propose Functional requirements	output
Propose Non-Functional Requirements	optionalInput

Ejemplo de uso: 12. Mostrar workproduct RiskList.

Cada usuario, al entrar al sistema, podría observar sus tareas, roles y productos de trabajo asignados. Por ejemplo, para un usuario concreto del proyecto se muestran sus workproducts, pudiendo acceder desde ahí a trabajar con ellos (actualizar su estado, extraer una copia, comitear una nueva versión, etc.).

Welcome Admin Person Help | My Info | **Todo** | Today | Logout

 **View User** add user 

users list : edit this user : edit preferences change password

Login Name: ignacioe	Birthday:
User Type: Default User	ICQ#:
Real Name: Ignacio E.	AOL Nick:
Company:	Email: ig@gm.es
Department:	Signature:
Phone:	
Home Phone:	
Mobile:	
Address:	




tabbed : flat

[Projects](#) |
 [Projects gantt](#) |
 [User Log](#) |
 [Permissions](#) |
 [Roles](#) |
 [EPF Possible Roles](#) |
 [EPF Specific Roles](#) |
 [EPF Tasks](#) |
 [EPF Workproducts](#)

Workproducts

List of all workproducts that has this user as responsible or as modifier in every project.

:: SimTraffic 0%

Order by	Workproduct name	As role	Workproduct Relation
	Configuration Management Report	QAInspector	Responsible
	Quality Assurance Report	QAInspector	Responsible
	OnePage Idea	SpokesMan	Responsible

Ejemplo de uso: 13. Pantalla de view user.

Por último, para conocer el estado del proyecto, se podría generar el informe modificado de dotProject que incluye datos sobre las nuevas funcionalidades de FVE. Así, además de gráficos que mostrarán los porcentajes de tareas o workproducts avanzados o completados, se generan informes generales que podrían haberse entregado en la asignatura como este y que habrían valorado de mejor manera las tareas de uno y otro usuario:

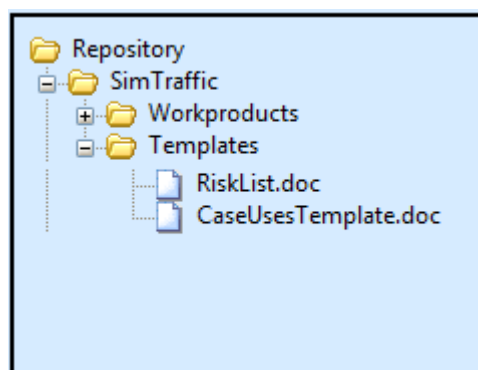
EPF Report

Users in Project	Tasks Primary	Tasks Additional	Workproducts Responsible	Workproducts Modifier	Possible Roles	Specific Roles
eugeniom	6	28	4	0	8	6
alfonsoc	2	23	1	0	6	6
alejandrov	14	24	3	0	7	5
inigof	3	22	1	0	7	6
ignacioe	11	26	3	0	10	9
nachom	3	13	1	0	7	6
nachoc	2	14	1	0	4	4
karena	0	11	0	0	2	2
raquelr	0	9	0	0	2	2
davida	2	14	0	0	5	4
davidg	2	10	1	0	5	4
juang	0	10	0	0	4	3
ulrichh	0	13	0	0	3	3
carlost	1	13	1	0	4	4
ivanr	0	13	0	0	3	3
pablof	4	24	1	0	5	5
sergiop	2	15	0	0	5	4
javierb	1	13	0	0	4	4
luiso	5	11	2	0	7	7
antoniog	1	10	1	0	4	3
Total:	59	316	20	0	14	14

Ejemplo de uso: 14. FVE stats report.

6.4 Visión y trabajo con el repositorio

Podrá consultarse en todo momento la información del repositorio y trabajar con él. En primera instancia se suben plantillas predefinidas al comienzo del proyecto, con el formato que tendrán los casos de uso y riesgos. Esto consigue de manera clara una mayor uniformidad en los contenidos que la que tuvimos inicialmente en ese proyecto, evitando esfuerzos posteriores en esta tarea. Se muestra el árbol del repositorio mostrando las plantillas cargadas.



Ejemplo de uso: 15. Árbol del repositorio.

Posteriormente pueden comitearse actualizaciones de los ficheros, eligiendo el archivo y el mensaje que se quiere dejar como commit

message. En este caso se trata de una actualización del formato de casos de uso de SimTraffic, que se produjo para pasar del formato existente en wiki al formato en word.

Commiting files

Choice the main folder where the commit will be introduced by clicking on the tree

Introduce here the commit message:

Ejemplo de uso: 16. Commit de archivos.

Para obtener una plantilla basta con ir a la página que muestra sus detalles y pulsar en la opción de descargar el fichero, siendo transparente la gestión de Subversion para el usuario. En nuestro caso, cuando desarrollamos ese proyecto en Ingeniería del Software, al no tener integrado el repositorio de código fuente en ninguna aplicación web, necesitábamos siempre de un ordenador con Eclipse para acceder a la documentación. En este caso, mediante acceso web por un navegador convencional, se pueden cargar y descargar los documentos de forma transparente (y mucho más rápida) para el usuario.

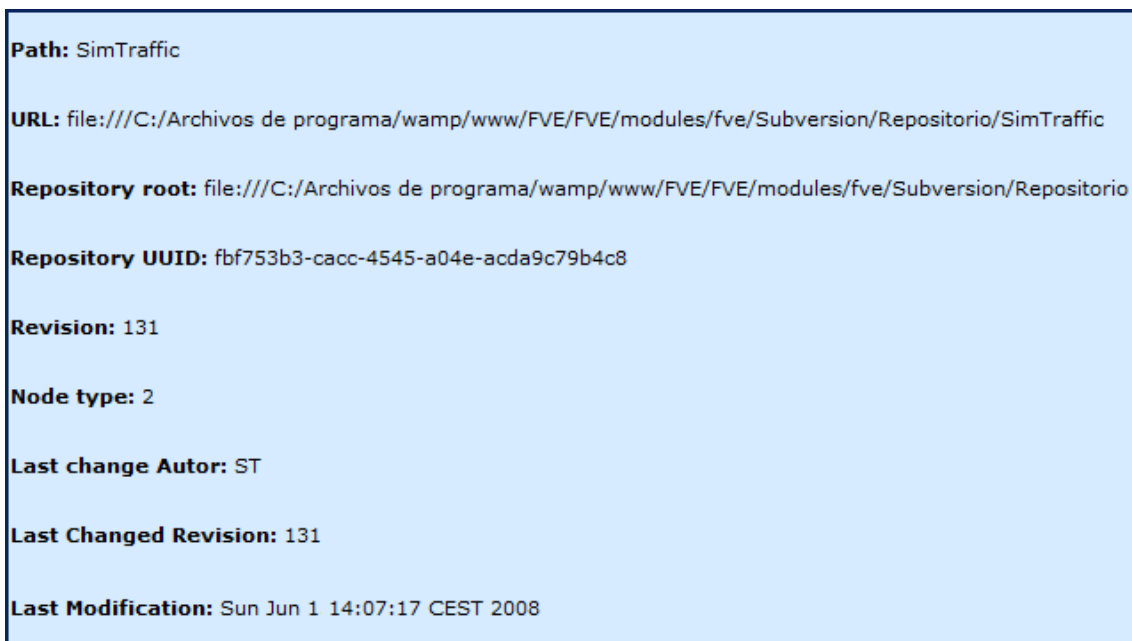
Show information about template RiskListIS

This is specific information about this template and his relations with workproducts, roles and users.

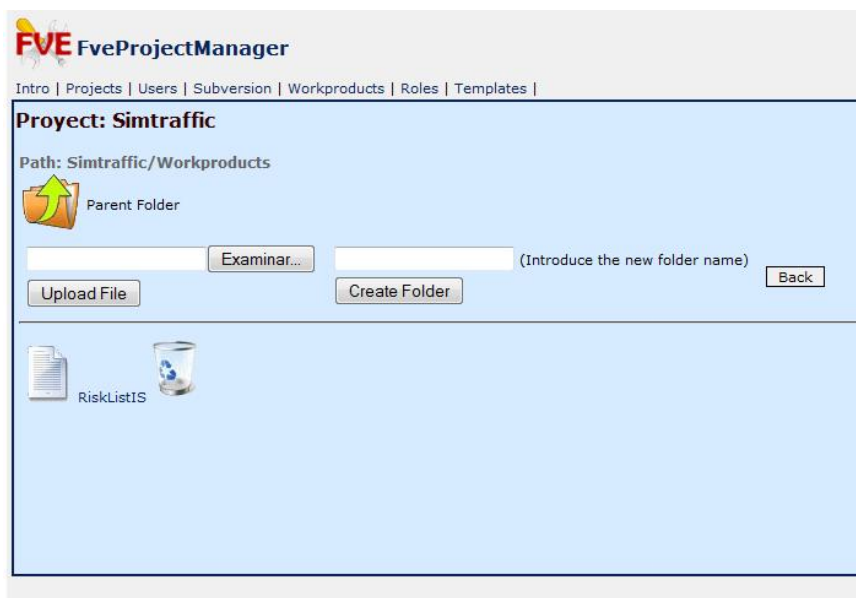
Id:	137
Name:	RiskListIS
Description:	Risk categorized by type and priority.
File:	Press here to link to the file or right click to download it
Part of project:	SimTraffic
Workproduct:	RiskList

Ejemplo de uso: 17. Mostrar Plantilla RiskListIS.

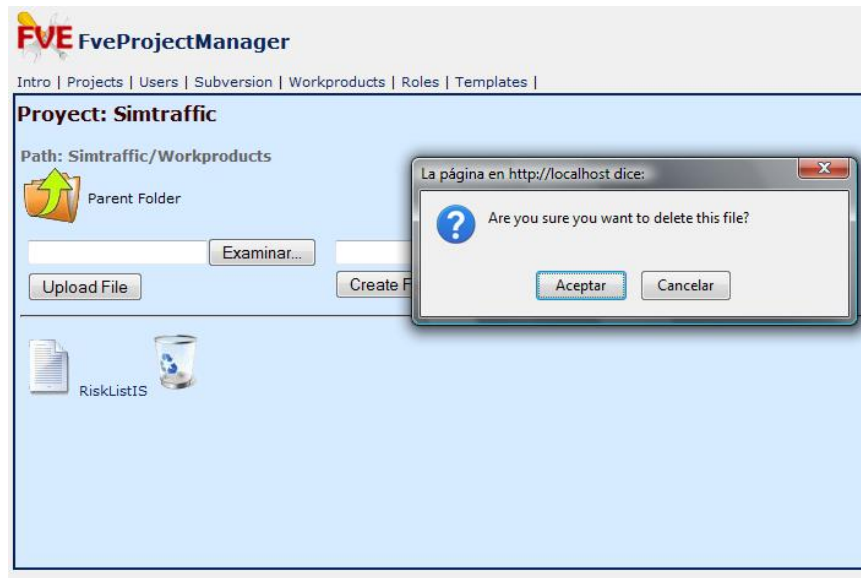
Así los usuarios pueden obtener la última versión y queda constancia en el repositorio de los últimos cambios realizados, tanto para una función informativa como para permitir, si fuera necesario, recuperar versiones anteriores.

**Ejemplo de uso: 18. Información del repositorio.**

A continuación podemos ver un ejemplo de borrado de workproduct. Realizarlo mediante el gestor "File Manager" que incorpora el paquete php de subversion es muy sencillo, simplemente debemos acceder a la carpeta de workproducts y pulsar el icono de papelera asociado al fichero elegido.

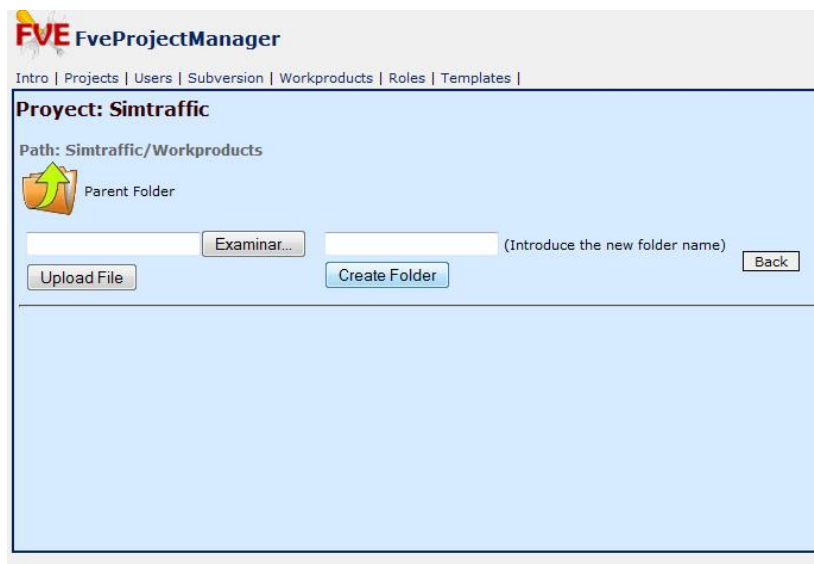
**Ejemplo de uso: 19. Ejemplo de workproduct del proyecto de Simtraffic visto como fichero físico en el FileManager.**

El sistema necesita de la confirmación por parte del usuario a la hora de eliminar un archivo para ofrecer así un mínimo de seguridad a la hora de realizar borrados:



Ejemplo de uso: 20. Ejemplo de confirmación de borrado de workproduct del proyecto de Simtraffic desde el gestor de archivos FileManager

En la siguiente imagen vemos el contenido definitivo del repositorio tras la eliminación del archivo. El modo de funcionamiento establecido es muy intuitivo y sencillo, se asimila al de un explorador de archivos de los sistemas operativos actuales, con la ventaja de estar trabajando directamente sobre el repositorio. Además permite incorporar al sistema con facilidad de manejo local todo el sistema de interacción con un repositorio.



Ejemplo de uso: 21. Situación posterior a la del borrado del único fichero de la carpeta de workproducts en el ejemplo de Simtraffic

Ésta es solo una visión general de las funcionalidades aplicadas a un proyecto concreto, mostrando algunas opciones de ejemplo, y cómo la herramienta habría sido de gran utilidad en nuestro proyecto (y lo habría sido en general en cualquier proyecto de esa asignatura). Para hacer un análisis completo de la funcionalidad implementada, los pasos a seguir para probarla y conocer todas las particularidades de la aplicación se remite al manual de usuario.

6.5. Otros ejemplos

Esta sección tiene como finalidad mostrar un ejemplo con detalle, pero se deja como referencia en esta sección las imágenes que muestran la importación de las principales metodologías definidas en EPF Composer (y descritas en el Anexo C.2) a nuestra aplicación.

6.5.1. OpenUp

	Pin	New Log	Work	P	Task Name	Task Creator	Assigned Users
		-	0%	↑	elaboration_phase_iteration	admin	(100%)
		-	0%		└─ manage_iteration	admin	(100%)
		-	0%		└─ identify_and_refine_requirements	admin	(100%)
		-	0%		└─ develop_architecture	admin	(100%)
		-	0%		└─ develop_solution	admin	(100%)
		-	0%		└─ Test Solution	admin	(100%)
		-	0%		└─ ongoing_tasks	admin	(100%)
		-	0%	↑	inception_phase_iteration	admin	(100%)
		-	0%		└─ initiate_project	admin	(100%)
		-	0%		└─ manage_iteration	admin	(100%)
		-	0%		└─ identify_and_refine_requirements	admin	(100%)
		-	0%		└─ agree_technical_approach	admin	(100%)
		-	0%	↑	transition_phase_iteration	admin	(100%)
		-	0%		└─ manage_iteration	admin	(100%)
		-	0%		└─ develop_solution	admin	(100%)
		-	0%		└─ test_solution	admin	(100%)
		-	0%		└─ ongoing_tasks	admin	(100%)
		-	0%	↑	construction_phase_iteration	admin	(100%)
		-	0%		└─ manage_iteration	admin	(100%)
		-	0%		└─ identify_and_refine_requirements	admin	(100%)
		-	0%		└─ develop_solution	admin	(100%)
		-	0%		└─ test_solution	admin	(100%)
		-	0%		└─ ongoing_tasks	admin	(100%)

Tareas definidas en OpenUp e importadas a FVE.

6.5.2. Scrum

tabbed : flat							
<div> Tasks Tasks (Inactive) Forums Gantt Chart Task Logs Events Files EPF Roles Workproducts Templa </div>							
	Pin	New Log	Work	P	Task Name	Task Creator	Assigned Users
		-	0%	↑	scrum_lifecycle	admin	(100%)
		-	0%		└─ Preparation Phase	admin	(100%)
		Log	0%		└─ plan_release	admin	Paco (100%)
		Log	0%		└─ initiate_product_backlog	admin	admin (100%) (+4)
		Log	0%		└─ Preparation work	admin	admin (100%)
		-	0%		└─ Sprint Phase	admin	(100%)
		-	0%		└─ Sprint (n)	admin	(100%)
		Log	0%		└─ plan_sprint	admin	berto (100%)
		Log	0%		└─ review_sprint	admin	berto (100%)
		Log	0%		└─ conduct_retrospective	admin	alex (100%)
		Log	0%		└─ manage_problems	admin	admin (100%)
		Log	0%		└─ conduct_scrum_daily_meeting	admin	admin (100%)
		Log	0%		└─ update_product_backlog	admin	Paco (100%)
		Log	0%		└─ Daily work	admin	(100%)
		Log	0%		└─ Sprint de release	admin	jorge (100%)
Key: =Future Task =Started and on time =Should have started =Overdue =Done Open : Clo							

Tareas definidas en Scrum e importadas a FVE.

6.5.3. eXtreme Programming

Tasks		Tasks (Inactive)		Forums		Gantt Chart		Task Logs		Events		Files		EPF Roles		Workproducts		Templ	
	Pin	New Log	Work	P	Task Name								Task Creator		Assigned Users				
		Log	0%		develop_xp_vision								admin		pepito (100%)				
		Log	0%		define_coding_standard								admin		jorge (100%)				
		Log	0%		estimate_task								admin		jorge (100%)				
		Log	0%		implement_spike								admin		jorge (100%)				
		Log	0%		refactor_code								admin		jorge (100%)				
		Log	0%		breakdown_story								admin		ignacio (100%)				
		Log	0%		estimate_user_story								admin		ignacio (100%)				
		Log	0%		integrate_system								admin		pepito (100%)				
		Log	0%		define_customer_test								admin		admin (100%)				
		Log	0%		report_acceptance_test_result								admin		admin (100%)				
		Log	0%		automate_acceptance_test								admin		pepito (100%)				
		Log	0%		run_acceptance_test								admin		pepito (100%)				
		Log	0%		setup_tester_environment								admin		pepito (100%)				
		Log	0%		adapt_and_improve_process								admin		admin (100%)				
		Log	0%		explain_process								admin		admin (100%)				
		Log	0%		improve_team_skills								admin		admin (100%)				
		Log	0%		keep_process_on_track								admin		admin (100%)				
		Log	0%		resolve_conflicts								admin		admin (100%)				
		Log	0%		track_story_completion								admin		admin (100%)				
		Log	0%		track_task_completion								admin		admin (100%)				
		Log	0%		define_release_plan								admin		sor (100%)				
		Log	0%		report_project_status								admin		sor (100%)				
		Log	0%		update_iteration_plan								admin		sor (100%)				
		Log	0%		update_release_plan								admin		sor (100%)				
		Log	0%		setup_programmer_environment								admin		admin (100%)				
		Log	0%		define_iteration_plan								admin		admin (100%)				
		Log	0%		write_user_story								admin		admin (100%)				
		Log	0%		write_code								admin		sor (100%)				

Tareas definidas en XP e importadas a FVE.

7. Conclusiones y trabajo futuro

El proyecto se ha centrado en construir una herramienta que aúne la flexibilidad en la definición y ejecución de procesos de desarrollo con un sistema de control de cambios. Estos objetivos han sido cubiertos tal y cómo habían sido definidos inicialmente, permitiendo que el resultado sea una herramienta a la que se pueda dar uso inmediato tras la realización del proyecto.

La herramienta ha sido probada fundamentalmente por los miembros de este proyecto y por el profesor responsable, aunque se ha proporcionado también a usuarios del proyecto SimTraffic del curso académico 2006/2007 [GrupoSimTraffic 07] (ejemplo de uso principal, consultar sección 6) para que evaluaran la diferencia en la planificación de un proyecto con esta herramienta y sin ella y favorecieran la evolución de la aplicación. Además, a la finalización de esta memoria se está dando el anuncio oficial en diferentes foros de desarrolladores, como el de dotProject, Subversion o EPF Composer. Es de esperar que el uso por parte de terceros haga mejorar más la herramienta.

El desarrollo del proyecto ha presentado dificultades en diferentes etapas: adentrarse con detalle en tantas herramientas estudiadas requería una gran cantidad de tiempo y una decisión equivocada en la elección de herramientas suponía una gran pérdida de esfuerzo en un trabajo que no formara parte de la aplicación final. También se unía a esto la complicación de trabajar con lenguajes nuevos en los que no se tenía experiencia e integrar más de uno en una misma aplicación. Por todo ello los avances no siempre fueron lineales respecto a los espacios cronológicos o esfuerzos realizados, y a medida que se avanzaba en la aplicación los progresos eran mayores. La labor previa a la implementación del proyecto, generando documentación sobre todos los aspectos y descubriéndose detalles significativos, fue en nuestra opinión un factor decisivo para que se hayan cumplido los objetivos propuestos.

7.1 Datos de la aplicación resultante

El proceso de desarrollo dio como resultado una nueva herramienta, llamada FVE Project Manager, que cumplía los objetivos propuestos y descritos en este documento. Junto al producto final generado se proporcionaron un manual de uso e instalación en inglés que se adjuntan en esta memoria.

La aplicación ha sido obtenida tras un proceso de implementación con una duración algo superior a los cuatro meses. A este respecto se muestra el contenido de la aplicación en líneas de código (lo cual incluye el código de dotProject, svnkit, librerías utilizadas, etc.), y las líneas de código que

corresponden a trabajo exclusivamente nuevo y realizado en el proyecto, empleando el software.

Listado de contenido detallado por lenguaje de la aplicación completa:

Como observación se puede destacar que el caracter multilenguaje de ciertas funcionalidades añadidas es el causante de que en la primera lectura de líneas de código aparezcan contenidos asociados a otros lenguajes, aunque se hace uso de sus implementaciones para PHP y Java.

http://cloc.sourceforge.net v 1.03 T=82.0 s (23.8 files/s, 6674.4 lines/s)						
Language	files	blank	comment	code	scale	3rd gen. equiv
PHP	709	23674	33499	119922 x	3.50 =	419727.00
HTML	265	11334	5502	96040 x	1.90 =	183996.00
Java	503	10279	26392	69042 x	1.36 =	93897.12
XML	51	462	256	60268 x	1.90 =	114509.20
Javascript	246	7635	10461	47318 x	1.48 =	70030.64
CSS	31	1026	656	5442 x	1.00 =	5442.00
Perl	18	707	1184	3513 x	4.00 =	14052.00
SQL	17	336	39	2251 x	2.29 =	5154.79
ColdFusion	23	379	660	1895 x	4.00 =	7580.00
JSP	34	172	267	1473 x	1.48 =	2180.04
ASP	15	267	319	1301 x	1.29 =	1678.29
Python	13	248	186	1038 x	4.20 =	4359.60
XSLT	3	81	21	364 x	1.90 =	691.60
DTD	7	26	36	154 x	1.90 =	292.60
ASP.Net	3	13	4	145 x	1.29 =	187.05
DOS Batch	7	26	0	86 x	0.63 =	54.18
Bourne Shell	5	30	0	70 x	3.81 =	266.70
SUM:	1950	56695	79482	411122 x	2.25 =	924098.81

CLOC Report: Líneas de código por lenguaje de la aplicación final.

Listado de los nuevos contenidos, suprimiendo del cálculo cualquier contenido no implementado desde el inicio.

El nuevo código fuente, calculado tras la limpieza de los contenidos no utilizados y las últimas refactorizaciones para reorganizar la información, presenta 18.000 líneas de código (ya suprimiendo la longitud de los ficheros xml con los que se ha trabajado, generados por EPF Composer y entregados con la aplicación).

http://cloc.sourceforge.net v 1.03 T=5.0 s (33.4 files/s, 13906.0 lines/s)						
Language	files	blank	comment	code	scale	3rd gen. equiv
XML	9	351	0	43580 x	1.90 =	82802.00
PHP	129	3739	2813	16015 x	3.50 =	56052.50
CSS	2	130	66	746 x	1.00 =	746.00
Java	7	176	209	559 x	1.36 =	760.24
JSP	13	80	39	547 x	1.48 =	809.56
SQL	2	69	20	216 x	2.29 =	494.64
HTML	1	7	4	110 x	1.90 =	209.00
Javascript	4	3	35	16 x	1.48 =	23.68
SUM:	167	4555	3186	61789 x	2.30 =	141897.62

CLOC Report: Líneas de código por lenguaje de la nueva funcionalidad.

Además, se muestra la actividad de la fase de elaboración y depuración mostrando la actividad en el sistema CVS de control de versiones empleado para alojar el código fuente.

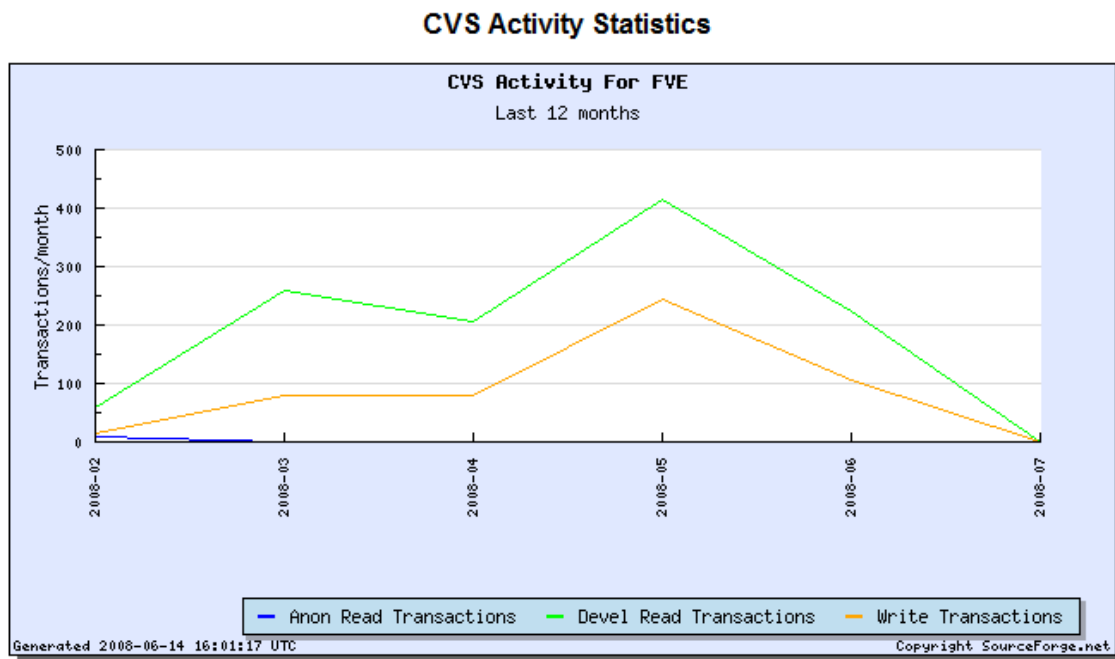


Gráfico de actividad del CVS del proyecto.

El producto generado está disponible para su descarga en <http://sourceforge.net/projects/fve/> desde la entrega final del proyecto.

7.2 Trabajo futuro

Pese a que la aplicación cubre las funcionalidades propuestas, el resultado final puede mejorarse y es factible ampliar sus funcionalidades. En primer lugar, una de las primeras mejoras podría ser reducir las dependencias creadas con la estructura base de dotProject, mejorando el concepto modular de FVE. No obstante, ciertas funcionalidades deberían concebirse de otra forma para poder cumplir con estos requisitos. Aparte de esto, se definen a continuación dos tipos de contenidos que podrían calificarse como trabajo futuro, el cual podría continuarse incluso con soporte y colaboración por parte de la comunidad de desarrolladores y colaboradores de dotProject si generase interés suficiente:

Aspectos que podrían ser mejorados de los contenidos implementados y que la arquitectura actual soportaría sin una gran carga de trabajo:

- *Ampliación de la exportación de los datos del proceso de desarrollo:* Los procesos de desarrollo de EPF Composer presentan una componente de documentación muy grande, presentando un gran número de atributos por cada entidad y diversas entidades que tienen el fin único de servir de

contenido de consulta. Por ello en la aplicación se ha optado por restringir la importación a las entidades más prácticas (como roles, productos de trabajo, tareas y plantillas) sin importar las definiciones de términos, listas de cuestiones sobre tareas, etc., y se ha incluido por cada elemento al menos el nombre y una breve descripción, pero no todos los campos posibles. No obstante, de cara a instanciar un proyecto a partir de un proceso de desarrollo, pueden existir usuarios que quieran transferir a dotProject toda esa información de EPF Composer y se podría ampliar el parser para que fuera así. Además, una vez incluida esta información, podrían incluirse opciones para el usuario para seleccionar con qué nivel de detalle quería importar el proceso de desarrollo y así actuar de una u otra forma de forma particularizada.

- *Ampliación de las operaciones disponibles ante un repositorio Subversion:*

El trabajo de integración de Subversion en dotProject ha supuesto mayores complicaciones estructurales que la importación de los procesos de desarrollo. Por ello se han conseguido implementar las opciones básicas que permiten visualizar el contenido del repositorio, introducir y extraer ficheros, ver la información general del repositorio y en general las opciones especificadas en los casos de uso relativos a Subversion, pero han quedado operaciones secundarias por implementar que serían necesarias si se quisiera emplear la herramienta como gestor de información de Subversion principal (como operaciones de diferencia entre archivos, obtención de versiones anteriores u otras opciones de administración de Subversion).

Aspectos que podrían incluirse pero se descartaron en la fase inicial por exceder el alcance del proyecto o han surgido en la fase final sin tiempo para implementarse:

- *Operaciones de soporte de aplicaciones:* Una de las posibles funciones que no cubren tampoco los gestores de proyecto es la configuración de herramientas en entornos locales. Así habría resultado también interesante realizar funcionalidad desde el gestor de proyectos encargada, por ejemplo, no sólo de descargar en la máquina local el código fuente de un repositorio, sino en arrancar el entorno de desarrollo y cargar los contenidos descargados automáticamente. También se valoraría por los usuarios de los gestores de proyectos, de acuerdo a lo escrito en los foros y listas de correo de estas herramientas, funciones que abrieran aplicaciones de escritorio como MsProject o MsOffice de acuerdo a las órdenes concretas del gestor web con contenidos específicos del proyecto seleccionados.

- *Mejora de entorno colaborativo del gestor de proyectos:* Uno de los aspectos que está más en auge en internet es el uso de sistemas web 2.0, destacando como cada vez está más implantado el uso de sistemas wiki o similares o cómo están cambiando las funcionalidades y aspecto de los foros que antes eran los elementos principales de intercambio de información. Una posible mejora en el entorno colaborativo de dotProject unida a este

módulo supondría llegar a más usuarios y aportar otro contenido que se considera uno de los puntos débiles de dotProject.

En definitiva, se ha conseguido llevar a cabo un proyecto completo de integración de diferentes herramientas, centralizando la información en un gestor de proyectos con funcionalidades no existentes entre las opciones libres existentes en la red. No obstante, este trabajo puede continuar en el futuro con la mejora de las funcionalidades y con la inclusión de otras relacionadas que complementarían el proyecto pero que requerirían de un nuevo proyecto para ser realizadas.

8. Bibliografía

[CE1 03]. Comunicación de la Comisión - El papel de las universidades en la Europa del conocimiento. Comisión Europea EAC.

<http://europa.eu/scadplus/leg/es/cha/c11067.htm>

[CE2 06]. Comunicación de la Comisión al Consejo y al Parlamento Europeo - Cumplir la agenda de modernización para las universidades - Educación, investigación e Innovación. Comisión Europea EAC.

<http://europa.eu/scadplus/leg/es/cha/c11089.htm>

[dP 06]. Reference from DotProject of article: Are Open Source software tools better than commercial?

<http://www.dotproject.net/index.php?name=News&file=article&sid=130>

[dPDocumentation1 08]. dotProject documentation (wiki). dotProject Team and Community.

<http://docs.dotproject.net/>

[dPDocumentation2 07] dotProject standards of code. dotProject Team and Community.

http://docs.dotproject.net/index.php/Coding_Standard

[EclipseFoundation1 06]. An Introduction to EPF Composer (Eclipse Foundation):

http://www.eclipse.org/epf/general/An_Introduction_to_EPF.zip

[EclipseFoundation2 05]. What's EPF (Eclipse Foundation):

http://www.eclipse.org/epf/community/Whats_EPF.ppt

[EPF1 08]. EPF Official Manual Documentation. Varios autores (trabajo colaborativo).

<http://epf.eclipse.org/>

[EPF2 08]. EPF Official Documentation: OpenUp. Varios autores (trabajo colaborativo).

<http://epf.eclipse.org/wikis/openup>

[EPF3 08]. EPF Official Documentation: Scrum. Varios autores (trabajo colaborativo).

<http://epf.eclipse.org/wikis/scrum>

[EPF4 08]. EPF Official Documentation: eXtreme Programming. Varios autores (trabajo colaborativo).

<http://epf.eclipse.org/wikis/xp/>

[Fish 07]. Better SCM Initiative. Shlomi Fish.

<http://better-scm.berlios.de/>

[Fowler 00]. Continuous Integration: Maintain a single source repository. Martin Fowler. Septiembre 2000.

<http://martinfowler.com/articles/continuousIntegration.html#MaintainASingleSourceRepository>

[GrupoSimTraffic 07]. Documentación de proyecto de Ingeniería del Software 2006/2007 SimTraffic: Simulador de tráfico vehicular. Ignacio Estirado Casado, Íñigo Fábregas Martínez, Alejandro Vilela Palacios y 17 autores más.

[Hughes 04]. SimpleXML article. Sterling Hughes.

<http://devzone.zend.com/node/view/id/688>

[M&T 04]. Are Open Source software tools better than commercial? (Methods&Tools)

<http://www.methodsandtools.com/dynpoll/oldpoll.php?OSQual2>

[MySQLHispano 99]. Transacciones en MySQL. MySQL Hispano. Enero 1999.

http://www.programacion.net/php/articulo/mysql_transacciones/

[OMG1 07]. Software & Systems Process Engineering Metamodel Specification, v2.0, Object Management Group:

<http://www.omg.org/docs/ptc/07-08-07.pdf>

[PHPManual1 07]. Subversion for PHP. PHP Manual.

<http://ca3.php.net/manual/es/intro.svn.php>

[PHPManual2 08]. XML Parser documentation. PHP official manual.

<http://es2.php.net/xml>

[SFDP 08]. Espacio en SourceForge de dotProject y estadísticas de descarga.

http://sourceforge.net/project/showfiles.php?group_id=21656

[SVNKit 07]. SVNKit Documentation: [Sub] Versioning for Java

<http://svnkit.com/documentation.html>

[Wikipedia1 08]. IBM Rational Method Composer. Wikipedia.

http://en.wikipedia.org/wiki/IBM_Rational_Method_Composer

Otras referencias indirectas:

[Álvarez 02]. Gestión de proyectos con Subversion. Juanjo Álvarez.

http://www.escomposlinux.org/fer_y_juanjo/index.php?pag=subversion.html

[Barrapunto 04]. Gestores de proyectos con interfaz web. Barrapunto. Septiembre 2004.

<http://softlibre.barrapunto.com/article.pl?sid=04/09/30/1131250>

[Collins-Sussman 06]. Version Control with Subversion. Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato.

<http://svnbook.red-bean.com/en/1.2/index.html>

[Colomer 07]. Gestores de proyectos Open Source. Rubén Colomer.

<http://www.online.com.es/2242/internet/gestores-de-proyectos-opensource/>

[DevioBlog 07]. Installing Trac on Windows. Devioblog.

<http://devio.wordpress.com/2007/12/03/installing-trac-on-windows/>

[dPTraining 08]. dotProject Training Materials and Courseware.

<http://training.dotproject.net/>

[IBM1 07]. An enterprise process framework defined and delivered with IBM Rational Method Composer. Febrero 2007.

<http://www.ibm.com/developerworks/rational/library/feb07/snyder/index.html>

[IBM2 08]. IBM Rational Method Composer – Features and Benefits (IBM):

<http://www-306.ibm.com/software/awdtools/rmc/features/index.html>

[Kobayashi 08]. La Ingeniería de Software en el desarrollo de software libre. Alfred Kobayashi Gutiérrez.

<http://linux.pucp.edu.pe/downloads/linuxweek2008/martes/Ingenier%C3%ADa%20de%20Software%20en%20el%20Desarrollo%20de%20Software%20Libre.pdf>

[OMG2 07]. Business Process Definition MetaModel (BPDM), Beta 1. *OMG Adopted Specification*.

<http://www.omg.org/docs/dtc/07-07-01.pdf>

[OpenSourceCMS 08]. OpenSourceCMS.

<http://www.opensourcecms.com/>

[Pressman 05]. Ingeniería del software. Roger S. Pressman. McGraw Hill. 2005.

[Talens-Oliag 07]. Gestión de proyectos informáticos con Software libre: Trac. Sergio Talens-Oliag. III Jornadas de Software Libre de Albacete. Mayo 2007.

<http://www.uv.es/~sto/charlas/GPICSL/GPICSL.html>

[Tigris1 08]. Subversion development.

<http://subversion.tigris.org/development.html>

[Tigris2 02]. Subversion design

<http://subversion.tigris.org/design.html>

[Wikipedia2 08]. Eclipse Process Framework (Wikipedia):

http://en.wikipedia.org/wiki/Eclipse_Process_Framework

Anexo A – Guía de estilo

La actual guía de estilo tiene como finalidad orientar al lector a conocer el estilo empleado en el actual documento y guiar a los desarrolladores a la hora de mantener la unicidad en todo el documento.

Los siguientes son los puntos utilizados para clasificar los contenidos de las secciones:

A.1 Primer punto

A.1.1 Primer inciso

A.1.1.1 Por si hiciera falta más

El tipo de fuente empleada es Verdana, con un tamaño de fuente 11 para el texto normal de los párrafos. Si se desea resaltar algún texto, se puede optar por la **negrita** o la *cursiva*.

En algunos puntos será necesario hacer referencia a código fuente. El formato empleado será como el que aparece a continuación. Utiliza fuente Courier New y tamaño de letra 10.

```
<?php
    $query = "SELECT projects.project_id, projects.project_name
FROM projects, fve_projects where
projects.project_id=fve_projects.project_id";
    //conectarBD(); //Con esto no iría
    $resultado = mysql_query($query) or die ('No puedo hacer el
query debido a : ' . mysql_error());

    while ($fila = mysql_fetch_assoc($resultado)){
        echo "<option value=\"{$fila[project_id]}\">
{$fila[project_name] (id:{$fila[project_id]}) </option>";
    }
?>
```

Si fuera necesario poner una **imagen** se puede hacer de la forma que se considere más adecuada en base al contenido, siempre y cuando se respeten los márgenes y no aparezca encima de texto o el pie o encabezado de la página. Incluirán un pie de foto.

La memoria está organizada en secciones, que se pueden consultar en cualquier página porque aparecen en el encabezado de la misma. En el pie de página aparece el número de página actual junto al nombre de la asignatura donde se desarrolla todo este trabajo.

Las referencias a otras obras o fuentes externas se realizarán mediante una llamada de este tipo [Fowler 00], conteniendo el autor de la obra y la fecha de publicación o última revisión, encontrando los datos de la obra completa en el apartado de bibliografía.

Anexo B. Trabajo relacionado con gestores de proyectos

Debido a no tener decidido desde el primer momento sobre qué gestor de proyectos implementar el proyecto, se realizó un estudio de gestores y una labor de introducción a las más importantes, haciendo especial énfasis en el gestor de proyectos elegido finalmente, dotProject.

B.1. Estudio de gestores de proyectos

El número de gestores de proyecto recopilados en un listado durante la primera fase de investigación fue muy alto: Superior a 20 aplicaciones. No obstante, se seleccionaron tras un primer examen los gestores de proyectos que debido a su naturaleza open source, su número de usuarios y sus características mejor se adaptaban al objetivo del proyecto. Entre estos, se realizó un examen más detallado, aportando como resultado las siguientes tablas en diciembre de 2007:

B.1.1. DotProject

<u>Propiedades</u>	<u>Descripción</u>
Nombre	DotProject
Fecha última versión (estable)	14 – Octubre – 2007
Nivel de desarrollo	Muy alto
Posibilidad de crear módulos	Sí, bien diseñado para ello (http://docs.dotproject.net/index.php/Programmer_Manual).
Comunidad	Foro con bastante información. Número de usuarios alto.
Open Source/Licencia	Sí/GPL
Lenguajes de prog.	PHP
Bases de datos:	MySQL por defecto, aunque acepta también Postgres y otras.
Descripción aspectos	El gestor de proyectos con más módulos realizados por

diferenciadores:	<p>usuarios “externos”.</p> <p>La aplicación con mayor comunidad de las vistas, con una wiki muy completa.</p> <p>Es el que incluye ya mayores aspectos relativos a la gestión de proyectos como diagramas de gantt o sistema de dependencias de tareas.</p>
Posibles riesgos	Se está pensando en hacer una versión nueva que cambie el funcionamiento interno de DotProject profundamente, tras sacar una versión en Octubre de 2007 donde el esfuerzo de depuración y solución de bugs está siendo excesivo debido a una estructura anticuada. Esto va a obligar a un cambio completo que se apunta desde los desarrolladores que obligará a refactorizar las aplicaciones que se hayan construido para las versiones anteriores.
Le falta	Su mayor defecto es que resultan muy limitadas las posibilidades de trabajo colaborativo. Tiene soporte de usuarios, pero estos no pueden comunicarse entre sí mediante mensajes internos. Presenta un sistema de foros, pero no otros módulos como wiki (al menos en los módulos principales). Además tiene soporte para archivos, pero no presenta complementos de integración de CVS y subversion.
Comercial	Existe soporte comercial.
Web	http://www.dotproject.net/

B.1.2. EGroupWare

<u>Propiedades</u>	<u>Descripción</u>
Nombre	EGroupWare
Fecha última versión (estable)	25 – Septiembre – 2007
Nivel de desarrollo	Alto, está activo actualmente.
Posibilidad de crear módulos	Sí, están organizadas las tareas en módulos, aunque es algo más complicado modificarlo que en DotProject.

Comunidad	Foro con bastante información. Número de usuarios alto.
Open Source/Licencia	Sí/GPL
Lenguajes de prog.	PHP
Bases de datos:	MySQL, PostgreSQL, MaxDB, Oracle no completamente.
Descripción aspectos diferenciadores:	<ul style="list-style-type: none"> - Más orientado a trabajo en grupo que otros como dotProject. - Tiene aspectos de CRM. - Wiki. - Interfaz más claro que en otros casos, visualmente más atractivo.
Posibles riesgos	<p>Parece necesitar en un aprendizaje mayor para hacer módulos sobre él, no está tan separada la creación de módulos del núcleo como en otros.</p> <p>Se adapta a muchos sistemas, por lo que puede tener demasiadas opciones que no tengan tanta utilidad en la aplicación a realizar, y ser excesivo para manejar algunos proyectos informáticos.</p>
Le falta	Mayor comunidad de usuarios.
Comercial	Ofrece soporte comercial, pero todo el programa se presenta de forma libre.
Web	http://www.egroupware.org

B.1.3. PhpCollab

<u>Propiedades</u>	<u>Descripción</u>
Nombre	PhpCollab
Fecha última versión (estable)	3 – Junio – 2005
Nivel de desarrollo	Bajo. Ha estado parado todo 2006 y parte de 2007, y ahora están desarrollando una nueva versión y

	buscando colaboración.
Posibilidad de crear módulos	En las versiones actuales, es muy complicado añadir módulos nuevos. Las experiencias encontradas por parte de los usuarios en este sentido no son positivas. No obstante, se apunta a una reestructuración completa para la siguiente versión en la que se solucionarán estos problemas.
Comunidad	Debido a que ha estado parado tanto tiempo, presenta menos actividad que en otras alternativas, aunque hay debates recientes en su foro.
Open Source/Licencia	Sí/GPL
Lenguajes de prog.	PHP
Bases de datos:	Mysql, PostgreSQL, Sql Server.
Descripción aspectos diferenciadores:	Tiene bastante recorrido ya, por lo que se supone un proyecto muy maduro y estable en versiones anteriores a la última.
Posibles riesgos	Está anticuada la versión actual, tanto en funciones de bases de datos como en modularidad. Aunque hay una nueva versión en marcha, no estará disponible una versión estable cuando se quiera comenzar la implementación de nuestra aplicación.
Le falta	Gestión de repositorio, y menos opciones que otros.
Comercial	Todo el programa se presenta de forma libre.
Web	http://www.php-collab.com/

B.1.4. PhpProjekt

<u>Propiedades</u>	<u>Descripción</u>
Nombre	PhpProjekt
Fecha última versión (estable)	20 – Diciembre -2006 (aunque el número de módulos que se han desarrollado desde entonces ha sido muy alto)
Nivel de desarrollo	Medio-Alto

Posibilidad de crear módulos	Sí
Comunidad	Media-Alta, foro activo.
Open Source/Licencia	Sí/GPL
Lenguajes de prog.	PHP
Bases de datos:	MySQL y otras.
Descripción aspectos diferenciadores:	- Apoyo de Google a esta aplicación OpenSource encuadrando tareas dentro de las iniciativas de Google "Summer of Code" (en este caso, "Summer of Project"). Los avances por ello han sido notorios en fechas recientes.
Posibles riesgos	- El programa es alemán, y la documentación parece que se escribe primero en alemán, lo cual puede suponer mayores problemas en la comprensión del código y la documentación (aunque tiene traducida la gran mayoría de la documentación).
Le falta	- Más orientada a trabajo en grupo que a gestión específica de proyectos informáticos. - No han sido probados todos los módulos, pero en la demo disponible se encuentra una interfaz menos clara que en otros proyectos, y parece que menos opciones.
Comercial	No
Web	http://www.phprojekt.com/

B.1.5. Open Workbench

<u>Propiedades</u>	<u>Descripción</u>
Nombre	Open Workbench
Fecha última versión (estable)	Septiembre de 2007.
Nivel de desarrollo	Alto

Posibilidad de crear módulos	No queda muy clara esta posibilidad.
Comunidad	Es un programa bastante usado de acuerdo al número de descargas (40000 última versión) que presenta y número de usuarios (100.000 indican en la web oficial).
Open Source/Licencia	Sí/GPL
Lenguajes de prog.	Java
Bases de datos:	MySQL y otras.
Descripción aspectos diferenciadores:	Es más parecido a Ms. Project y muy diferente a las otras alternativas. Diagramas Gantt, Pert, etc.
Posibles riesgos	Es un programa de escritorio, con las ventajas y sobre todo desventajas que esto conlleva.
Le falta	El soporte web del resto de gestores estudiados.
Comercial	Existen opciones de soporte, aunque la funcionalidad principal es libre. Aunque hay que destacar que para usarse como desarrollador se utiliza Microsoft Visual Studio 2003 que requiere de licencia.
Web	http://www.openworkbench.org/

B.1.6. Web2project

<u>Propiedades</u>	<u>Descripción</u>
Nombre	Web2Project
Fecha última versión (estable)	Aun no tiene, está en fase de desarrollo.
Nivel de desarrollo	Desconocido.
Posibilidad de crear módulos	En principio sí.
Comunidad	No tiene aunque bastantes usuarios se han

	interesado por el proyecto en su etapa inicial.
Open Source/Licencia	Sí
Lenguajes de prog.	PHP
Bases de datos:	Desconocido
Características:	<ul style="list-style-type: none"> - Infraestructura web capaz de manejar proyectos, compañías, departamentos, y usuarios. - Integración modular de las características necesarias en cada proyecto, configurando entornos de negocio. - Mantendrá y actualizará el módulo de gestión de proyectos y dependencias entre tareas ya presente en dotProject. - Permisos basados en roles y más.
Descripción aspectos diferenciadores:	Ha surgido como una evolución de dotProject, ante los problemas internos que ha habido en el otro proyecto. Va a utilizar el framework de dotProject, luego el código inicial será similar. De hecho, algunos de los desarrolladores han participado activamente en el código de dotProject.
Posibles riesgos	Al ser un proyecto nuevo, le falta comunidad y demostrar lo que puede aportar ante tantas soluciones como hay ya en este momento.
Le falta	Comprobar como evoluciona en los próximos meses, y como responden los usuarios.
Comercial	No
Web	http://www.web2project.net/

B.1.7. ActiveCollab

<u>Propiedades</u>	<u>Descripción</u>
Nombre	ActiveCollab
Fecha última versión (estable)	En continua actualización.

Nivel de desarrollo	Alto
Posibilidad de crear módulos	Sí, estructura de paquetes, parece que tenida en cuenta desde el comienzo del desarrollo del programa.
Comunidad	Menor a otras
Open Source/Licencia	En un principio lo parecía, pero cambiaron la licencia para usos de más de 5 usuarios. En la versión 0.7.1, HPL (básicamente GPLv2 con soporte ASP)
Lenguajes de prog.	PHP5
Bases de datos:	MySQL y otras.
Descripción aspectos diferenciadores:	Presenta un interfaz más claro que otras alternativas, y parece que su arquitectura es más clara.
Posibles riesgos	No está tan claro como en otras opciones su enfoque Open Source.
Le falta	Mayor estabilidad.
Comercial	Surgió como una versión libre de BaseCamp (http://www.basecamp.com/) por su parecido, pero posteriormente convirtieron la solución en gratuita para una gestión de 1 a 5 proyectos y de pago para el resto de usos.
Web	http://www.activecollab.com/

B.1.8. Gestor de proyecto a bajo nivel, con gestión y seguimiento de control de versiones (Subversion): Trac

<u>Propiedades</u>	<u>Descripción</u>
Nombre	Trac
Fecha última versión (estable)	0.10.4, en octubre de 2007.
Nivel de desarrollo	Medio
Posibilidad de crear módulos	Sí, aunque en un principio este no sería el uso que necesitaríamos en la aplicación a construir.

Comunidad	Alta (es un módulo muy utilizado y en aumento).
Open Source/Licencia	Si/BSD (versiones anteriores GPL)
Lenguajes de prog.	Php / Python
Bases de datos:	SQLite por defecto, pero pueden ponerse otras como MySQL.
Características:	<ul style="list-style-type: none"> - Sistema de tickets (para almacenar bugs en general, pero disponibles para otras tareas). - Visor de Subversion - Wiki integrada, facilitando la información de desarrollo. - Integrable con otras aplicaciones de integración continua, otros visores, etc.
Descripción aspectos diferenciadores:	Entorno colaborativo, y mayor cercanía al repositorio de control de versiones como eje del proyecto.
Posibles riesgos	Centrarse en integrar una herramienta como ésta y no destinar por ello suficientes recursos en el enfoque de gestión de procesos software (mucho más alejado de la implementación)
Le falta	Sería una aplicación que unir a la principal, con la que se quieren lograr los objetivos: Por si sola no se considera un gestor de proyectos como el que se quiere desarrollar.
Comercial	No
Web	http://trac.edgewall.org/

B.2. Introducción y uso general de dotProject

DotProject es una aplicación PHP que en su versión 2.1.1 tiene un tamaño aproximado de 100.000 líneas de código (excluyendo los módulos opcionales desarrollados que no están incluidos en su núcleo). Su código fuente está estructurado en módulos y separado por funcionalidades y tipos de archivos. La mayor parte de la aplicación está comentada, por lo que pueden seguirse sus estándares de programación y comprobar cómo han sido implementados sus contenidos, aunque debido a su estructura claramente modular no se requiere un conocimiento tan profundo de la arquitectura interna de dotProject para desarrollar módulos en esta aplicación. Además hay que reseñar que el nivel de complejidad de ciertas secciones del código es bastante alto y, aunque se usan clases, no presenta una estructura que se base en el paradigma de orientación a objetos por lo que no es sencillo adentrarse en su arquitectura.



dotProject logo.

La instalación de dotProject en un equipo es bastante sencilla y no plantea ningún problema recurrente. Se requiere tener activo un servidor web con soporte para php y un servidor MySQL, en el que habrá que definir una base de datos que utilizará dotProject. Una vez instalado dotProject, éste realizará una serie de operaciones de auto-configuración y estará listo para ejecutarse.

El medio que utiliza dotProject por defecto para almacenar información es una base de datos MySQL (aunque soporta otras). El programa realiza diferentes consultas sobre esta base de datos y va creando la información que el usuario introduce vía web. Existen un total de 75 tablas, aunque las funciones principales que ofrecen estas tablas están enfocadas al almacenamiento de entidades: Compañías, proyectos, tareas, usuarios y roles (aunque con una funcionalidad muy limitada y sin permisos asociados). Además una parte importante de las tablas se dedican a configurar opciones de dotProject y sus módulos y a establecer permisos específicos.

El uso habitual de dotProject sería la creación de proyectos, asociados a compañías, indicando datos sobre el estado del proyecto, y la creación posterior de recursos y tareas con usuarios asociados a éstos. Para ello debería moverse entre las opciones del menú superior que presenta la aplicación. Una vez generadas las entidades existe un “comportamiento” esperado en la aplicación, de modo que cada función View de una entidad presenta siempre un cuadro con sus datos en la parte superior, y un menú con pestañas en la parte inferior con los listados de elementos relacionados. Además, siempre aparecen botones de creación de las entidades en el menú superior derecha. Este tipo de comportamiento, que mostramos en la siguiente imagen remarcando las secciones habituales en recuadros verdes, será tenido en cuenta a la hora de crear nuevas funcionalidades para que el modo de uso de la aplicación sea también el “esperado” por los usuarios habituales de dotProject.

Welcome User Person Help | My Info | **Todo** | Today | Logout

View ELEMENT
element list : edit this element

new element new attribute

Details

Values

Attributes

Description

tabbed : flat

Selected Others

P	Name	Owner	Others

Interfaz: Comportamiento esperado de dotProject.

Anexo C. Trabajo relacionado con EPF

C.1. Uso de EPF Composer

Debido a la importancia de esta herramienta en el proyecto se ha hecho un uso completo de la herramienta y se ha profundizado en todas las fases de la construcción de un proceso de desarrollo. Se detallan en este apartado las operaciones realizadas para hacer uso de esta herramienta, desde un punto de vista general, y que fueron elaboradas para posteriormente desarrollar una metodología completa en el apartado 2.2.6.

C.1.1. Configuración de entorno para un nuevo proceso de desarrollo.

- a) Crear un plugin nuevo con la opción File -> New -> Method Plugin, completando el resto de valores que se piden como nombre, descripción, autores y librerías en que se apoya (pudiendo incluir OpenUp u otra metodología definida para poder tomar contenidos concretos de dicha librería y no tener que crear el proceso completo desde cero).
- b) Asociar una Configuración al plugin: File -> New -> Method Configuration.
- c) Asociar una vista a la configuración anterior. Pestaña Views, y seleccionar una vista nueva o reutilizar una vista de otra librería si no se va a crear un nuevo estilo para visualizar la información.

C.1.2. Creación de un proceso de desarrollo vacío.

- a) Acceder en el menú situado a la derecha en la aplicación hasta la ruta: Plugin -> Processes -> Delivery Processes. Después botón derecho, New -> Delivery Process.
- b) Rellenar información general y en detalle del proceso.
- c) Se permite añadir más configuraciones válidas para este proceso, además de la configuración creada anteriormente e indicada por defecto.

C.1.3. Creación de uno o más conjuntos de roles (Roleset), para poder crear posteriormente uno o más roles asociados.

- a) Si es la primera vez que se crea un elemento en el proceso de desarrollo, crear un paquete para el contenido (Content Package), haciendo click con el

botón derecho sobre Content Packages, Opción New, y posteriormente Content Package.

- b) Rellenar datos generales del Content Package.
- c) Crear un conjunto de roles, pinchando en la opción Standard Categories del nuevo Content Package, haciendo click en el botón derecho sobre RoleSet, y seleccionando la opción New RoleSet.
- d) Rellenar documentación general.
- e) Crear un rol concreto, haciendo click sobre la opción Roles del Content Package creado, botón derecho opción New y después Role.
- f) Rellenar los datos del nuevo rol en la ventana que aparece: Nombre, descripción, habilidades, sinónimos, etc.
- g) Asignar el nuevo rol al Roleset creado anteriormente, en la pestaña Categories y pulsando sobre la opción Add, seleccionando el adecuado.

C.1.4. Creación de uno o más tipos de productos de trabajo (Workproduct Kinds), para poder crear posteriormente uno o más Work Products.

- a) Crear un tipo de workproducts, pinchando en la opción Standard Categories del Content Package creado, haciendo click en el botón derecho sobre WorkProduct Kind, y seleccionando la opción New WorkProduct Kind.
- b) Rellenar documentación general.
- c) Crear un producto de trabajo concreto, haciendo click sobre la opción Workproducts del Content Package creado, botón derecho opción New y seleccionando después Workproduct.
- d) Rellenar los datos del nuevo workproduct en la ventana que aparece: Nombre, descripción, objetivo, impacto de no tener, etc.
- e) Asignar al Roleset creado anteriormente, en la pestaña Categories y pulsando sobre la opción Add, obteniendo el adecuado de una lista.

C.1.5. Creación de tareas y jerarquías en un proceso de desarrollo: Iteraciones, Fases, Actividades y Tareas.

- a) Seleccionamos el proceso de desarrollo creado, y al abrirlo encontramos diferentes pestañas, eligiendo Work Breakdown Structure para la creación

de tareas. Como las tareas van a ser específicas del proceso de desarrollo, pueden crearse aquí y no como tareas generales de un Content Package.

b) Botón derecho sobre una estructura de la tabla de tareas (inicialmente solo estará la tarea superior que corresponde al propio proceso de desarrollo), opción new child, y seleccionar el elemento que quiere crearse entre los diferentes tipos de tareas posibles: Task descriptor, Activity, Phase, Iteration, Milestone.

c) Si existe algún módulo de una librería estándar (desde una simple tarea a una iteración completa) que quiera ser transferida al nuevo proceso de desarrollo se buscará en la ventana de Configuración la tarea descrita en su subdirectorio, y se copiará desde ahí, o se hará una asociación a la tarea, creando una tarea nueva y seleccionando en Method Task la opción Link Method Element.

d) Sobre cada tarea se podrá de nuevo rellenar información complementaria.

C.1.6. Asociación de entidades.

Al crear cada entidad se pueden realizar asociaciones que serán las que realmente definirán el grafo de la aplicación y los distintos diagramas del proceso de desarrollo. Así podrán seleccionarse:

a) En un rol, en la pestaña Workproducts, los productos de trabajo de los que este rol es responsable, y los que puede modificar por estar asociados a una tarea en la que este rol participa.

b) En una tarea, en la pestaña Roles, se podrá seleccionar cuál será el rol encargado principal (Primary Performer) y cuáles los roles adicionales (Additional Performer). En la pestaña Workproducts, se podrán definir los workproducts que tienen relación con la tarea, especificando si son entradas obligatorias al sistema (mandatory input), entradas opcionales (optional input) o salidas (output).

c) En un workproduct, en la pestaña Guidance, si tiene algún tipo de guía asociada (plantilla sobre la que empezar a trabajar o ejemplo de cómo realizar este producto de trabajo).

C.1.7. Publicación Web del proceso de desarrollo

Esta opción también presente en la solución de IBM permite crear una web html o j2ee con toda la información del proceso desarrollado. Para conseguirlo hay que seguir los siguientes pasos:

- a) Pinchar en la barra de menú superior sobre Configurations, y seleccionar la opción Publish.
- b) Seleccionar la configuración que quiere publicarse, y pulsar sobre Next.
- c) Seleccionar el proceso de desarrollo (o la librería completa) sobre el que quiere realizarse el sitio web, y pulsar sobre Next.
- d) Especificar las opciones concretas (título de la web, información que se quiere incluir, diagramas seleccionados, etc.) y pulsar sobre Next.
- e) Seleccionar finalmente si se desea un sitio web html o un fichero .war para ser desplegado en un entorno J2EE y pulsar sobre Finish, obteniendo el sitio web en la carpeta especificada.

C.1.8. Exportación de datos

Además de poder publicar la información introducida en el proceso de desarrollo creado, existen algunas opciones adicionales de exportación de datos, que fueron detalladamente analizadas debido a la importancia que tenía en nuestro proyecto tratar esta información fuera de EPF Composer. Para exportar los datos, se debe hacer clic en el menú superior en File, y seleccionar la opción Export, obteniendo las siguientes opciones:

- a) Library Configuration: Extrae la información completa de la librería de procesos de desarrollo actual, para ser importada en otra instancia de EPF Composer.
- b) Method Plugin: Similar a la anterior opción, pero se exportará únicamente el plugin que especifique el usuario.
- c) MsProject: Esta opción permite en principio exportar la información definida en EPF Composer a un fichero XML que puede leer la aplicación de gestión de Proyectos MsProject. No obstante, el estudio de esta alternativa mostró que la única información exportada era la relativa a las tareas, y perdiendo la mayor parte de la documentación introducida o qué tipo de tarea era cada una importada. Además, la asociación que se realizaba de los roles era convertirlos directamente en usuarios, una aproximación simple para los objetivos del proyecto y para respetar la propia estructura de EPF Composer.

La ventaja inherente, no obstante, de esta opción, era la posibilidad de aprovechar al menos algunos de los datos introducidos en una herramienta líder en el sector, y la posibilidad por su popularidad de que existieran plugins que relacionaron la información de MsProject con otros gestores de proyectos web Open Source sobre los que se pretendía

trabajar, por lo que se realizaron pruebas detalladas para analizar la estructura XML del fichero.

d) XML: Pese a que la opción anterior generaba un fichero XML, esta es la verdadera opción de exportación a XML porque en este caso si se exporta toda la información textual introducida en EPF (es decir, toda la información excepto los ficheros asociados como plantillas, imágenes asociadas, etc). Este fichero sigue un Schema publicado por EPF Composer bastante más complicado que el fichero XML de MsProject, aunque se asume en la primera fase del proyecto que pese a ser la opción más ambiciosa sería la más apropiada para tratar datos fuera de EPF Composer y no perder muchos datos relevantes en la conversión a otras herramientas.

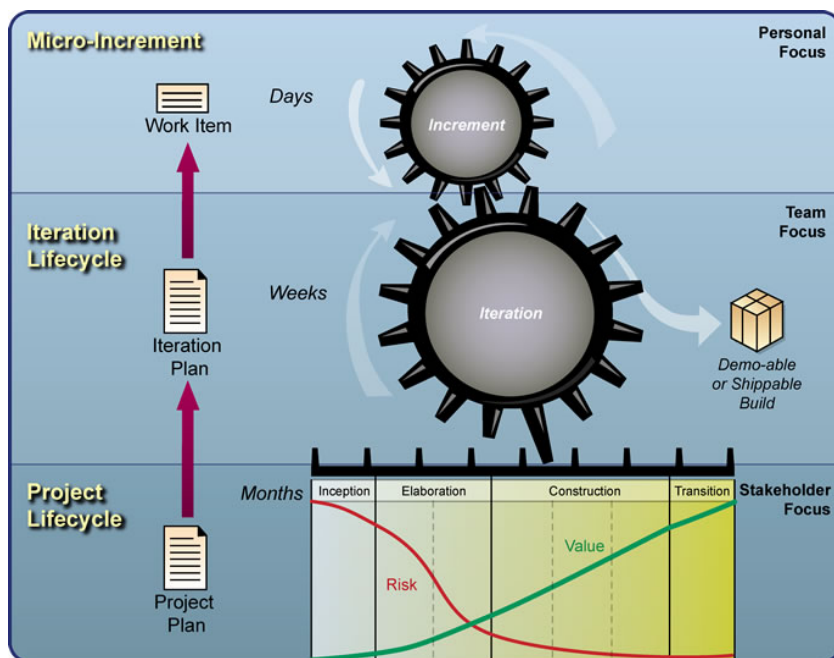
C.2. Metodologías definidas en EPF y utilizadas en el proyecto.

Como se ha citado ya en la descripción de EPF Composer, los esfuerzos de Eclipse Process Framework no están solo destinados a conseguir una herramienta que trabaje con procesos de desarrollo, sino realizar un esfuerzo en la definición de nuevas metodologías o en adaptar metodologías existentes en este entorno. Debido a lo reciente de esta herramienta y los esfuerzos que conlleva cada una de estas metodologías solo se encuentran definidas 4 metodologías: OpenUp, OpenUp Basic, Scrum y eXtremme Programming. Estas metodologías (descritas a continuación), unidas a las que se crearon manualmente como la definida en el apartado 2.2.6, se han utilizado como ficheros de prueba en el proyecto siendo el primer resultado visible de la importación de datos de EPF Composer a un gestor de proyectos web.

C.2.1. OpenUp y OpenUp Basic

Se trata de la metodología más cercana al modelo unificado, y presenta una versión para equipos reducidos denominada OpenUp Basic. Esta metodología se define como mínima, completa y extensible, y mediante estos conceptos trata de asociar las ideas de las metodologías ágiles con los conceptos tradicionales de planificación del proyecto.

Supone la metodología más completa definida en EPF Composer: Delimita Roles, Workproducts, Tareas y Guías, con el máximo nivel de detalle (indicando pasos de cada tarea, objetivos, etc.). De hecho, de cara a una posible exportación a un gestor de proyectos, no toda la información será importada por carecer de utilidad una vez implantado definitivamente en un grupo y en un proyecto.

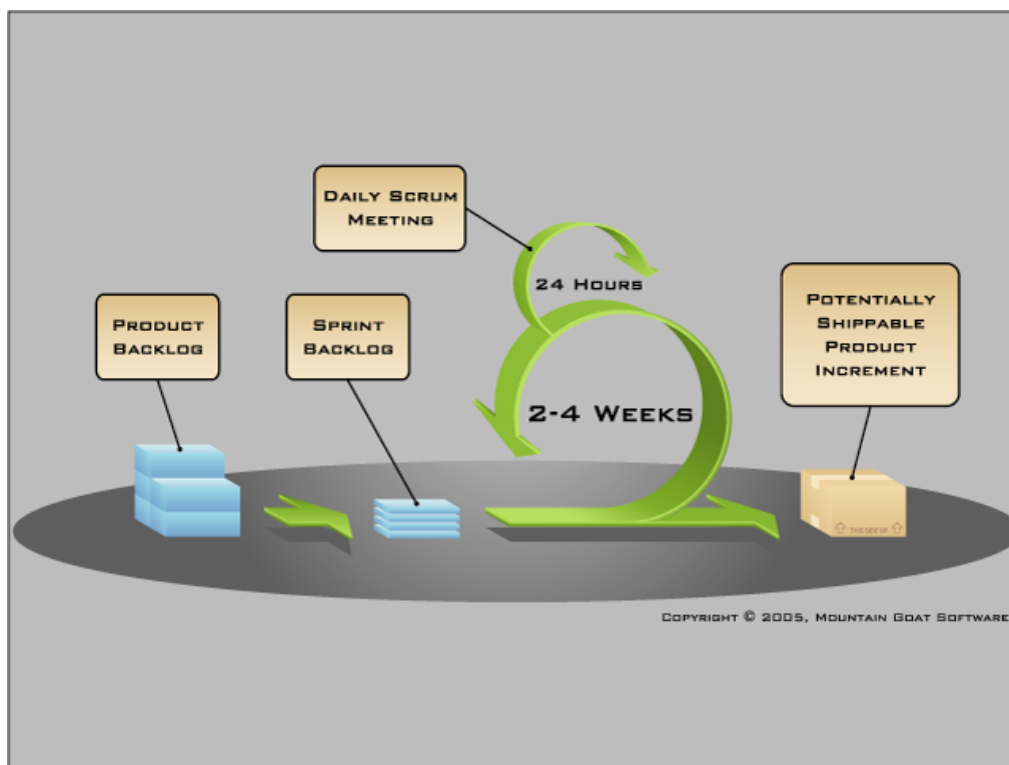


[EPF2 08]. Diagrama de funcionamiento OpenUp.

C.2.2. Scrum

Scrum es una metodología existente antes de la aparición de EPF Composer que se ha definido posteriormente en este entorno para proporcionar más ejemplos desarrollados con la herramienta. Scrum trata de simplificar las tareas y se apoya en el compromiso de los equipos, que realizan trabajos concretos en un periodo de unos 20 días aportando nuevas versiones, en la fase conocida como Sprint.

Esta metodología, tal y como está definida en EPF Composer, es menos completa que OpenUp y no trae plantillas por defecto, aunque están definidos los roles, productos de trabajo y tareas con las jerarquías necesarias y es suficiente para realizar la planificación de un proyecto para los conocedores de la metodología.

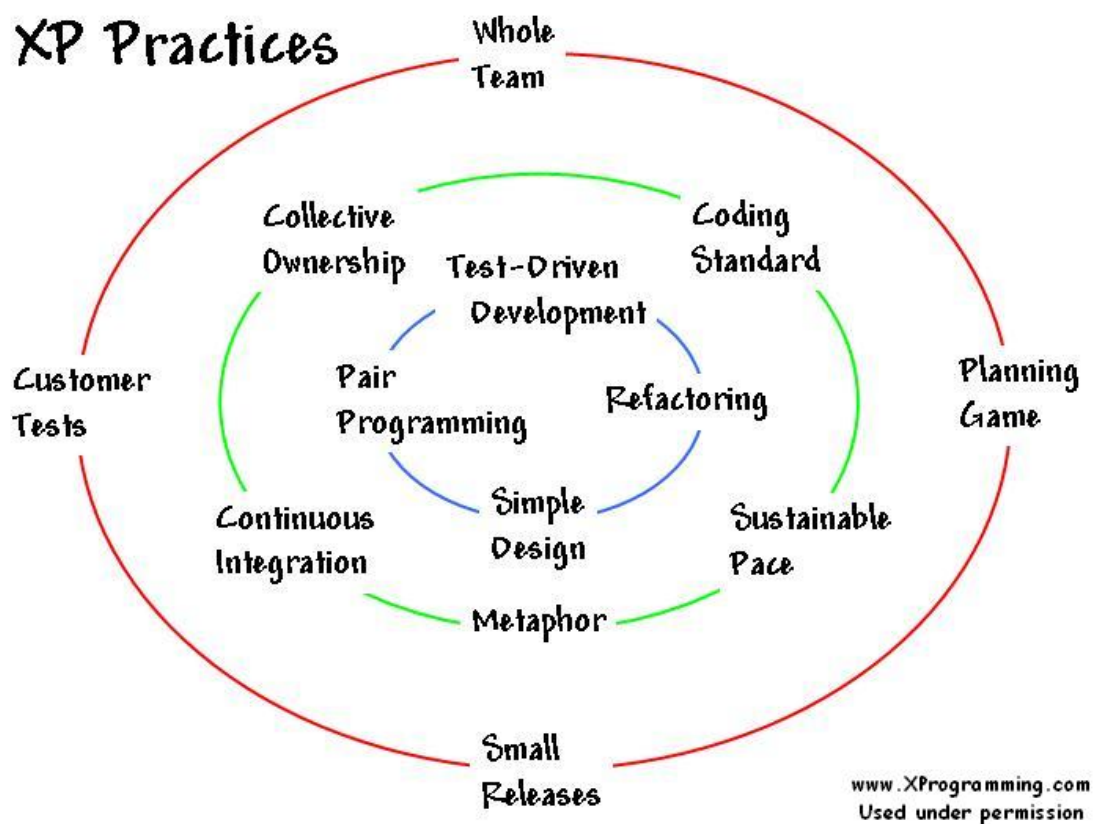


[EPF3 08]. Diagrama de funcionamiento Scrum.

C.2.3. eXtreme Programming

eXtreme Programming es la última metodología (también existente al margen del proyecto Eclipse Process Framework) que ha sido convertida a un proceso bajo la estructura de EPF Composer. Con un lema conocido como "Make it Easy" (Hazlo fácil), esta metodología se ha convertido en una de las más usadas del mercado y se basa en un trabajo estructurado para equipos pequeños y medios que trata de evitar los esfuerzos innecesarios.

Esta metodología no incluye una jerarquía completa de tareas padres e hijas aún en la definición en EPF Composer, pero sí incluye los productos de trabajo, roles y tareas que implica eXtreme Programming.



[EPF4 08]. Diagrama de funcionamiento eXtreme Programming.

Anexo D – Trabajo relacionado con Subversion

D.1 – Instalación y uso de Subversion

D.1.1 - Instalación

Subversion es un programa que no requiere otros programas para funcionar, y además es multiplataforma, por lo que su instalación no es excesivamente complicada. Se ha instalado Subversion en Linux y Windows, concretamente la versión 1.4.5 de Agosto de 2007, sin que hayan aparecido problemas de importancia (al menos en la configuración por defecto).

D.1.2 - Creación y uso de un repositorio: Guía rápida

Una vez instalada la aplicación, se han probado los comandos habituales del repositorio por línea de comandos (ya que es posible y más cómodo utilizar un interfaz gráfico para hacerlo, pero la forma de hacerlo en el proyecto estaría más cercana a ésta). En esta guía rápida sobre el uso de un repositorio, se destacan los siguientes comandos habituales:

- *Creación de un repositorio:*

```
svnadmin create /path/to/repo
```

- *Importar un proyecto a un repositorio (inicialmente, si se parte de un código fuente existente):*

```
svn import /path/to/project /path/to/repo -m 'Import inicial'
```

Hay que apuntar, no obstante, que este comando supondrá que éste sea el estado de la revisión 0, de modo que si se quieren mantener las versiones de un código que estuviera en un repositorio de otro tipo habría que realizar alguna conversión previa (por ejemplo, si partimos de un código en un servidor CVS, se puede utilizar cvs2svn para no perder todo el historial).

- *Añadir un archivo o un directorio nuevo a un repositorio:*

```
svn add /path/to/newfile
```

- *Eliminar del repositorio un fichero:*

```
svn delete /path/to/file
```

- *Obtener una copia local del código existente en el repositorio:*

```
svn checkout /path/to/repo
```

- *Posteriormente el ciclo básico de trabajo en Subversion será:*

- Actualizar copia de trabajo:

```
svn update
```

- Hacer los cambios pertinentes (añadir y borrar ficheros):

```
svn add  
svn delete
```

- Examinar mis cambios:

```
svn status  
svn diff
```

- Fusionar cambios cuando se tienen 2 versiones distintas de un mismo fichero modificadas por dos usuarios, y se quieren reunir todos los cambios en uno:

```
svn merge
```

- Enviar los cambios al repositorio:

```
svn commit
```

- *Hacer una copia de seguridad del repositorio.*

```
svnadmin dump nombre repositorio > ficheroBackup
```

- *Configuración de permisos.*

Mediante un fichero externo auth, o mediante módulo de seguridad de Apache, que debe configurarse. En el fichero externo authz se escribirán los usuarios y los permisos por carpetas, anotando para cada ruta del repositorio un listado completo como:

```
[/path/concreteProject]  
* =  
user1 = rw  
user2 = r
```

D.2 Estudio de librerías de gestión SVN

Para integrar Subversion en una aplicación se valoraron diferentes alternativas:

D.2.1 - Librerías de Subversion para PHP

Esta alternativa presentaba la ventaja de que el estudio paralelo sobre gestores de proyectos apuntaba que PHP sería uno de los lenguajes implicados en el proyecto. Sin embargo, la solución fue descartada por el estado de las librerías, ya que se advertía en el manual oficial de PHP [PHPManual11 07] de que su uso era simplemente experimental en el momento actual, carecía de documentación suficiente, y presentaba pocas funciones auxiliares que no cubrían completamente los objetivos del proyecto.

D.2.2 - USVN

Se trata de una nueva solución para PHP, y es más una aplicación completa que una herramienta intermedia como se buscaba inicialmente. Se estudió la herramienta ya que gestionaba los permisos del repositorio y diferentes proyectos, de forma bastante sencilla, y de nuevo bajo el lenguaje deseado. No obstante, no aportaba implementaciones de operaciones comunes de subversion alejadas de esos objetivos, como commit, update, etc., aspecto necesario en la aplicación. Además, presentaba numerosas dependencias la gestión del repositorio con otras funcionalidades de la aplicación, por lo que reutilizarlo resultaría más complicado que en otras opciones.

D.2.3 - SVNKit

SVNKit, antes conocido como JSVN o "Java Subversion Library", es una herramienta empleada cuando se quiere acceder a un repositorio para consultar sus datos o modificarlos desde una aplicación Java. Es una opción que abstrae la complicación de trabajar directamente con Subversion y de tener que crear interfaces y funciones lógicas nuevas para cada tarea habitual en un repositorio.

Existe abundante documentación, y numerosos ejemplos con código fuente que hace ciertas operaciones habituales para los repositorios (añadir archivos, hacer commit sobre ficheros existentes (o directorios), eliminar archivos, etc.).

La librería es utilizada por una gran cantidad de proyectos importantes que trabajan de alguna forma con Subversion y que incluyen programación en Java. Por ejemplo, utilizan SVNKIT los plugins disponibles para trabajar con Eclipse (Subversive / Subclipse), el sistema de Integración Continua Hudson o visores del repositorio Subversion como SmartSvn o SVN Web Client entre otros [SVNKit 07]. Esto supondrá que se

tratará de código muy probado y que no provocará fallos inesperados si se elige como tecnología con la que trabajar durante el proyecto.

Anexo E – Prototipos

Previo al proceso de implementación de la herramienta se realizó una fase de estudio de los requisitos y herramientas disponibles que se muestran a lo largo de esta memoria. No obstante, durante esta fase también se realizaron tareas de prototipado de la aplicación y se generaron productos de prueba con código fuente PHP y HTML. En este apartado se muestran las funcionalidades de esos prototipos aunque finalmente no formaran parte de la entrega final del proyecto.

E.1 Prototipo de gestión de tareas y dependencias

E.1.1. Introducción e instalación del prototipo

Para realizar el prototipo de dotProject, hemos realizado un nuevo módulo con el nombre “prueba”. El archivo de instalación de nuestro módulo (setup.php) no realiza ninguna operación, pero está estructurado como una plantilla por si se desea añadir alguna operación en especial.

Una vez instalado como cualquier módulo de dotProject, sobre la pantalla principal de dotProject aparecerá nuestro módulo, “Prueba”. Si hacemos click sobre él, nos redirigirá a la pantalla principal del módulo en la que se muestra un pequeño texto, una imagen y un applet de Java.

Dentro de esta pantalla del módulo, observaremos arriba tres links, cada uno dirigirá a una página del módulo con un objetivo concreto. A continuación se muestra qué hace cada uno. Lo que hará la aplicación será cargar en función del link que se elija una página *.php* que contendrá los datos.

E.1.2 - Página principal

Esta página está contenida en el archivo *main.php*. Como se menciona anteriormente, la página muestra un ejemplo de código html con etiquetas CSS, la inclusión de una imagen y la aparición de un sencillo applet de Java que muestra un texto y un gráfico que surgen aleatoriamente dentro del área destinada al applet. La fuente del applet es un archivo *jar* que apenas ocupa 3KB ya que únicamente contiene los *class* del programa que la máquina virtual del equipo del cliente ejecutará.

E.1.3 - Interacción con proyectos

Esta página está contenida en el archivo *projects.php*. El objetivo de esta sección es la interacción con la estructura básica de información que utiliza dotProject, es decir, poder realizar operaciones como crear proyectos o tareas pero desde nuestro propio módulo. De esta forma, conseguimos aprender cómo realizar estas operaciones que de cara a nuestro proyecto serán necesarias.

Básicamente lo que permitimos es:

- Creación de proyectos: mediante un formulario se pueden crear nuevos proyectos.
- Creación de tareas: mediante un primer formulario, el usuario elige dentro de qué proyecto desea crear la tarea. Una vez elegida, en el segundo formulario elegirá los datos de la tarea y podrá elegir entre las tareas que hay en el proyecto elegido cuáles son de las que dependerá. Mediante dos botones se pasan tareas de una caja de texto a otra. Esto es un ejemplo de código Javascript, que permite modificación de formularios sin necesidad de recargar la página. Los proyectos diferentes, las tareas de los proyectos...etc. se obtienen a partir de consultas a la base de datos.

Cuando el usuario crea un proyecto o una tarea, se llevan a cabo por debajo las consultas SQL pertinentes y se genera la información deseada que se puede consultar desde los módulos estándar de dotProject. Las tablas que se utilizan en esta página son: **companies**, **projects**, **tasks** y **user_tasks**. Una vez creados los datos, se muestra por pantalla al usuario un mensaje de confirmación en la parte superior.

E.1.4 - Tareas avanzadas

El fin de esta página es la creación de tareas pero con la diferencia de que introducimos nueva información asociada a ellas con la que nuestro módulo podrá trabajar. En nuestro caso, asociamos a la tarea el papel de un proceso que podría tener cabida dentro de un flujo de trabajo. Para ello, creamos dos tablas nuevas en la base de datos de dotProject: **prueba_procesos_base** que contiene los procesos disponibles y **prueba_procesos_con_tareas** que relaciona las tareas con los procesos. La creación de estas tablas se podría haber realizado en el archivo de configuración del módulo, en *setup.php*.

El formulario es prácticamente análogo al de creación de tareas del apartado "interacción con proyectos", con la salvedad de que permite asociar la tarea a un proceso de los que hay disponibles en la base de datos.

Tanto en este apartado como en el anterior, a la hora de crear una tarea, se realiza una pequeña labor de parseo a raíz de que las tareas que el usuario a elegido como dependencias pasan a la página web de inserción como una cadena de texto en la que aparecen los identificadores de las tareas separados por comas. Esta cadena se genera por Javascript, ya que existe una pequeña limitación de php a la hora de utilizar formularios con cajas de selección múltiples.

El código del prototipo está comentado en su mayoría y su comprensión no resulta demasiado difícil.

E.2 Prototipo de interfaz de usuario

E.2.1 - Introducción

El objetivo de este prototipo de interfaz gráfico es poder tener una idea del resultado final que se quiere obtener de la aplicación e ir definiendo más concretamente todas las funcionalidades que se quieren incluir. Una visión general de lo que se quiere implementar ayuda mucho a ver la utilidad de cada implementación de los requisitos propuestos.

Pese a limitación funcional del prototipo desarrollado, se puede obtener una idea de las tecnologías que necesitaríamos utilizar para hacer que nuestra aplicación final funcionara como se ha planeado.

A continuación se encuentran varias secciones, cada una de las cuales explicará un aspecto relevante del prototipo de interfaz gráfico.

E.2.2 - Instalación

La tecnología empleada en el desarrollo de este prototipo ha sido PHP y MySQL. En primera instancia llevamos a cabo la instalación de dotProject sobre un servidor web contratado para realizar las pruebas. Este proceso requiere de la existencia de un servidor MySQL con una base de datos disponible. Una vez se cumpla este requisito, únicamente es necesario copiar dotProject al servidor y ejecutar el script de configuración, que nos pedirá los datos de acceso a la base de datos. DotProject se auto configurará y estará listo para utilizarse.

Una vez instalado dotProject, llevamos a cabo el proceso de instalación de nuestro módulo prototipo. DotProject tiene un sistema de ampliación de funcionalidades basado en módulos. Desarrollamos nuestro prototipo basándonos en las normas impuestas por dotProject para la inclusión de nuevos módulos en su motor de funcionamiento. Estas reglas son las siguientes:

- Crear una carpeta que contenga todo el contenido del módulo.
- Copiar esta carpeta a la carpeta **/modules** de dotProject.
- Configurar el archivo **setup.php**, que contiene una serie de variables y clases que dotProject necesita para la inclusión del módulo en su núcleo. La configuración de nuestro módulo fue muy sencilla, pero este sistema tiene mucha potencia y permite la ejecución de scripts que ejecuten, por ejemplo, la creación de tablas dentro de una base de datos.
- Desde dotProject, ir a la pestaña de administración y dentro del apartado "modules" instalar el nuevo módulo.
- En la parte superior de dotProject, junto al resto de módulos, aparecerá el nuestro, con el nombre InterfazUsuario.

E.2.3 - Estructura y funcionamiento

La estructura base que hemos seguido en nuestra interfaz es la de una página principal que contiene un menú con todas las opciones en la parte superior. La parte inferior de esta página cambia en función de la opción elegida en el menú superior. Esto se realiza todo desde un archivo `index.php` que cambia los contenidos a mostrar dependiendo de la variable que reciba en la barra de navegación (variables GET) del navegador. De esta forma, se reutiliza mucho código y es muy fácil la creación/modificación de las diferentes secciones.

Este es el aspecto que muestra la barra superior de la página que permanece siempre disponible:



Prototipo 2: Gestión de Workflows.

Gracias al sistema de hojas de estilo CSS utilizado en dotProject, nos fue muy sencillo adaptar el nuevo módulo al *estilo* utilizado por dotProject. Revisamos todas las clases definidas en el archivo CSS y vimos dónde hacía uso de ellas dotProject. Nosotros seguimos estas pautas y logramos que el nuevo módulo no desentonara con el estilo del resto de contenidos de la página.

Para la utilización del prototipo, creamos sobre dotProject un proyecto con varias tareas para poder relacionarlo de forma más cómoda con el interfaz. En la versión final del programa queremos que la relación entre los proyectos y tareas de dotProject y los de nuestro módulo sea muy estrecha y no tenga ningún tipo de problema de incompatibilidad.

A continuación explicaremos con mayor detalle qué realiza cada apartado del prototipo.

Intro

Este apartado tiene como único fin introducir los objetivos del módulo prototipo.

Proyectos

Será el pilar central de nuestro futuro módulo. Tendrá todas las herramientas para la creación de proyectos basados en workflows. Además, permitirá comprobar el estado en detalle de todos los proyectos activos para el usuario que esté conectado.

Si hacemos click sobre un proyecto, se mostrará toda la información relativa a él.

Información de proyecto

Cuando entremos en la pantalla de información de proyecto nos daremos cuenta de que es muy similar a la que utiliza dotProject, sin embargo, en este caso la información contenida estará relacionada además con la gestión de flujos de trabajo. Se verá información como el coste estimado del proyecto en líneas de código y en personas/mes (que se calcula en base a todas las tareas contenidas en el proyecto) o permitirá lanzar asistentes para concertar reuniones con los trabajadores del proyecto o resúmenes escritos del estado actual del proyecto.

Esta pantalla nos mostrará también la información de los elementos que contiene el proyecto, como tareas, casos de uso o requisitos. Estos dos últimos son exclusivos de nuestro módulo.

Tareas

Esta sección nos permitirá organizar todas las tareas que aparecen en los proyectos gestionados por nuestro módulo. Si pinchamos sobre una de ellas, pasaremos a la página de información de tarea.

Información de tarea

Gracias a esta página, podemos conocer toda la información relativa a una tarea. Es similar a la pantalla de información de tareas que proporciona dotProject, sin embargo en nuestro módulo esta pantalla contendrá información más específica de la tarea, relacionada con el sistema de workflows. Hallaremos información relativa a los puestos de trabajo requeridos con el rol de cada uno así como el personal asignado a cada puesto.

Tarea: Crear prototipos SDK

Detalles		Personal	
Proyecto:	Mod Half Life 2	Rol	Usuario
Tarea:	Crear prototipos SDK	Diseñador	nacho
Propietario:	admin	Desarrollador	alex
Priority:	normal	Desarrollador	ifigo
Progress:	0%	Desarrollador	luis
Tiempo Trabajado:	0	Desarrollador	-
		Tester	luis
Fechas y objetivos		Dependencias	
Fecha inicio :	10/12/2007 01:00 pm	Estudiar SDK	
Fecha fin :	11/12/2007 05:00 pm		
Duración estimada :	12 hours	Tareas que dependen de ésta	
Coste y estimaciones		none	
Líneas de código (estim.):	1200	Artefactos a generar	
Esfuerzo (estim.):	0.5 personas/mes	Tipo	Nombre artefacto
Más información		Estado	
Casos de uso relacionados:	2	Código	Prototipo funcional Source
Personal total:	5 personas	Doc	Walkthrough de los pasos seguidos con el SDK
Puestos (actual/óptimo):	5 / 6 (Luis 2 puestos)	Doc	Lista de problemas encontrados
Fase:	Construcción		
Proceso Unificado Padre :	Prototipado	Descripción	
		Hacer prototipos de diferentes funcionalidades que queremos utilizar en el Mod utilizando los conocimientos aprendidos con el SDK.	

Prototipo 2: Visualización de tarea.

También tendremos información de coste y estimaciones o el proceso unificado que engloba la tarea. Podremos asignar una serie de artefactos que deseamos que se generen al final de la tarea, como documentación o código fuente.

De cara a los nuevos “recursos” que utilizamos en nuestro módulo, podemos relacionar la tarea con una serie de casos de uso. Igualmente, podemos asignar recursos a una tarea. Estos recursos son los que se pueden definir en el módulo “Resources” de dotProject (por defecto este módulo se encuentra desactivado).

Todo este esquema de funcionamiento respeta la estructura base de dotProject, como fechas de inicio/fin o dependencias. Podríamos acceder a la información de nuestras tareas desde el propio dotProject aunque no encontraríamos toda la información nueva, para ello sería preciso entrar en nuestro módulo.

Usuarios y roles

Pese a que dotProject ofrece soporte tanto para usuario como para roles, su importancia se reduce únicamente a nivel de administración. En nuestro módulo, pretendemos asignar a los usuario una serie de roles en función de las tareas para las que están capacitados. El jefe de un proyecto podrá definir una tarea y especificar cuántas personas deben trabajar en ella y qué papel deben desempeñar. Por ejemplo, para una tarea de revisiones el jefe de proyecto puede definir que necesita ingenieros de pruebas y documentadores. Un jefe de grupo podría elegir a personas de su cuadrilla para que cumplan con los puestos requeridos.

Rol	Usuario
Diseñador	nacho
Desarrollador	alex
Desarrollador	iñigo
Desarrollador	luis
Desarrollador	-
Tester	luis

Prototipo 2: Listado de roles y usuarios

Esta página de nuestro módulo, permite la gestión de los usuarios así como de los roles que estos pueden desempeñar.

Al acceder a “Usuarios y Roles” veremos todos los usuarios existentes, junto con información relevante como su nombre o el número de tareas y proyectos en los que está trabajando. Si hacemos click sobre su nombre, accederemos a la pantalla de información de usuario.

Información de usuario

En esta página encontraremos toda la información relativa a un usuario en cuestión. Como en el resto de elementos, se respeta el formato adquirido por dotProject relativo a la información del usuario. Sobre esta información, se añade otra nueva de utilidad en nuestro módulo, como roles que puede desempeñar o usuarios de los que es responsable. Se establece una jerarquía entre usuarios de manera que cualquier usuario tiene un único responsable y puede ser responsable de varios usuarios.

Aparte de esta información, también encontraremos información detallada sobre las tareas en las que está trabajando, conociendo el proyecto en el que se encuentran enmarcadas. Hallaremos igualmente información sobre los casos de uso y requisitos que ha definido el usuario en cuestión.

Casos de uso

Como su nombre indica, en este apartado se listarán todos los casos de uso que se han definido en todos los proyectos del sistema. Si queremos ver la información ampliada de uno de ellos, debemos hacer click sobre su nombre.

Información de caso de uso

Al hacer click sobre un caso de uso en nuestro módulo, se accederá a la siguiente pantalla de información. En ella se verá el caso de uso definido con toda su información. Al tratarse de un elemento propio de nuestro módulo, no existirá ninguna referencia a él desde dotProject. Entre la información que podemos encontrar dentro del caso de uso encontramos sus objetos, los actores que intervienen, la secuencia de su utilización o los activadores que lo ponen en marcha.

Requisitos

Número	Tipo	Nombre	Importancia	Estado	Descripción
 Sistemas Informáticos :: Mod Half Life 2 33%					
1	Funcional	Ofrecer 5 modos de dificultad	Media	Desconocido	El juego debe de tener 5 modos de dificultad que el jugador elegirá...
2	Funcional	Modo historia con 12 niveles	Muy alta	En trámite	En el modo de historia, el jugador irá avanzando a través de diferentes...
3	Funcional	Arsenal de 8 armas	Alta	Cumplido	El jugador tendrá a su disposición 8 armas distintas, cada una ...
4	No Funcional	FPS > 30	Alta	Desconocido	Mantener una tasa de refresco de imagen superior a los 30 FPS.
5	No Funcional	Ratón y teclado	Muy Alta	Cumplido	Permitir el control mediante el teclado y el ratón y todas las combi...
6	No Funcional	PC Gama media	Media	En trámite	El juego debe poder funcionar en un PC de gama media que tenga...

Prototipo 2: Listado de Requisitos.

La última página de nuestro módulo es la de requisitos. Mostrará información de todos los requisitos definidos dentro de los proyectos existentes. Al hacer click sobre el nombre del requisito entraremos en el modo de información del requisito, que se detalla a continuación.

Información de requisito

La siguiente pantalla muestra información sobre un requisito en concreto y se muestra al hacer click sobre el nombre de un requisito dentro de nuestro módulo. La información que se muestra es la típica dentro del estándar de desarrollo de procesos y contiene información tal como la descripción del requisito, su importancia o su estado actual.

Identificación	
Número:	2
Tipo:	Funcional
Nombre:	Modo historia con 12 niveles
Descripción	
En el modo de historia, el jugador irá avanzando a través de diferentes niveles hasta lograr descubrir la trama. Para que el juego tenga suficiente peso y logre una gran inmersión en la partida, el número de niveles debe ser al menos de 12.	
Creador	alex
Importancia	
Muy alta	Un número de niveles muy bajo puede hacer que el jugador supere el juego en poco tiempo, dando menos calidad al producto. La historia, debido a su trama se calcula que necesitará de 12 niveles para su completa adaptación al juego.
Estado	
En trámite	Aún se están desarrollando los niveles.
Asuntos pendientes	
-	

Prototipo 2: Información de un requisito.

Anexo F – Manual de instalación

A continuación se incluye el manual de instalación que se incluye en formato HTML en nuestra aplicación. Es posible que algunas cosas no se vean de igual forma que en el manual debido a la conversión del formato HTML al del presente documento.

El objetivo de este manual es ayudar a los usuarios de la aplicación a instalar el programa en sus ordenadores.

F.1. Introduction

Thanks for choosing FVE Project Manager! This manual will guide you through all the necessary steps in order to install dotProject with FVE Project Manager functionalities. Please, follow the next points and you will be working with this application sooner than you think.

F.2. Installing the necessary services

If you want to install dotProject with FVE Project Manager, you need the next services installed in your computer:

- Apache web server
- Tomcat server
- MySQL service (it can be located anywhere, whether you can access it)
- Subversion service(only if you want to use subversion features over FVE)

If you don't have Apache nor MySQL installed, [xampp](#) could be very useful to you for getting web server and database services easily.

F.3. Installing the application

Web and MySQL Installation

Once you have these services ready, download the FVE compressed file into your file system. You have to unzip this file where you want to install the application. Remember that this folder will have to be accessible from your apache server.

Now use your favourite web browser to go to your localhost dotproject install dir. For example, <http://localhost/dotProject>. Once you reach this page, as this is the first time you run the application, the next message will arise:

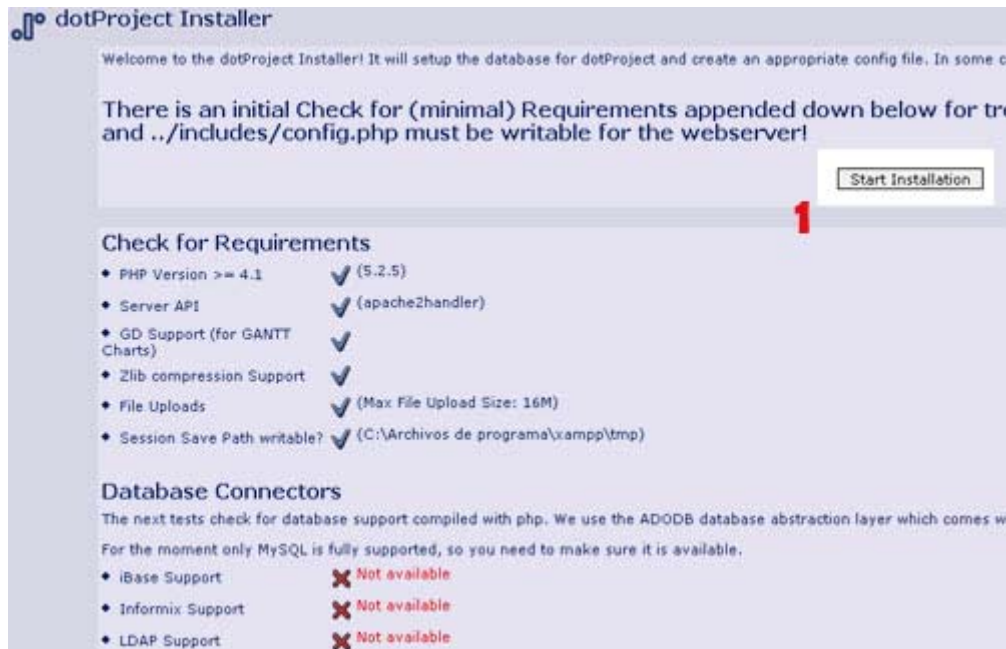
Fatal Error. You haven't created a config file yet.

[Click Here To Start Installation and Create One!](#) (forwarded in 5 sec.)

Error: Don't worry, it's not so fatal

This is normal, because there is no config file. DotProject will help you to create a new config file according to your current system settings. Just press "Click Here To Start Installation and Create One!" or wait for 5 seconds.

After this, you will see a page where you can check if your computer currently passes the needed requirement to make dotProject with FVE work. Check all the topics and if everything is ok, press the "Start Installation" button to jump to the next installation step.



This is the dotproject initial check page

Don't worry if all the ticks in this page are not checked, that's because there are many things that are optional. The next page you'll see is the database configuration page, that looks like this:

dotProject Installer

Database Settings

Database Server Type *Note - currently only MySQL is known to work correctly* MySQL - Recommended

Database Host Name

Database Name

Database User Name

Database User Password

Use Persistent Connection? ☐

Drop Existing Database? ☐ If checked, existing Data will be lost!

Download existing Data (Recommended)

Download a XML Schema File containing all Tables for the database entered above by clicking on the Button labeled 'Download XM'. This file can be used with the Backup module to restore a previous system. Depending on database size and system environment, process can take some time.

PLEASE CHECK THE RECEIVED FILE IMMEDIATELY FOR CONTENT AND CONSISTENCY AS ERROR MESSAGES ARE PRINTED INTO "

Database connection configuration page

As you can see, there are several fields that you have to fill:

- Database Host Name: this is where you have your MySQL database running.
- Database Name: this is the name of the database you want to use. If you haven't created one, do it through a MySQL prompt by introducing the following command: **create database dotProject;** This will create a database into your MySQL server with 'dotProject' name.
- Database User Name: the name of the user that will access to your dotProject database.
- Database User Password: the password of the user that will access to your dotProject database.

Once you have filled these fields, press the "install db & write cfg" button to set up your database with all the necessary tables used by dotProject.

If everything went right, your database will be created and you will be able to access to your new dotProject deployment. If you access now to your dotProject url, you will see the login page. Log in as admin with passwd password. This is the default user dotProject creates when it firstly initialize. Change its password as soon as possible.

The first time you enter into dotproject, an error will occur (*Table 'dotproject.fve_project_and_workers' doesn't exist*). This is normal, because the current version of dotProject tries to use database elements that are not yet installed. To do so, we need FVE project manager to install them. To do this, we have to install FVE module. To do this, follow the next steps.

FVE module Installation

The installation of this module is mandatory, because when you install it, it sets your database up in order to work with new EPF information. It also includes web pages to help you managing your EPF projects.

To install this module, go to System Administration module (in the upper part of the dotProject page) and then press the modules section press "View modules". You will access to the modules page, where all the available modules of the system are listed. Notice the "fve" module. Now, it's uninstalled. Press the "install" text to install it.

help	active	core	1.0.0	help	up.jpg	hidden	12
FVE: EPF Importer And Subversion Manager	disabled remove	user	1.0.0	FVE: Additional Info	moduleLogo.jpg	hidden	13
history	install						

The module is installed but not activated yet

In the previous image, you can see the installed module. But it is not activated yet. To do this, press the "disabled" text to activate it. Now your module will be activated and you will be able to use it in dotProject.

If you want to see the extra features of this module, press to "hidden" text to allow dotProject to show it in the main module bar (at the top of the screen). This image shows fve module installed and visible:

help	active	core	1.0.0	help	up.jpg	hidden	12
FVE: EPF Importer And Subversion Manager	active remove	user	1.0.0	FVE: Additional Info	moduleLogo.jpg	visible	13
history	install						

Your FVE module is active and ready to use!

With all the previous steps, you have accomplished the web and database configuration of your dotProject with FVE application. Now you have to configure all your Subversion repository configuration to allow your projects work with online repositories. Please go to the next section of this manual to learn how to set up your Subversion configuration.

Subversion installation

If you want to install subversion FVE Project Manager module, you will need to acquire (these features are free) those services first:

- Install subversion service
- Tomcat service

When you have the last version of these programs follow these instructions:

1. Install subversion service

2. Install tomcat service

Here it's important to remember the url where tomcat publish the web pages, by default this path is XXX.XXX.XXX.XXX:8080 where X represents your external ip. In this case you just have to remember the port access 8080.

3. Activate tomcat service

Activate it and make sure that is running accessing it main web administration page, normally emplaced in <http://localhost:8080/>.

4. Deploy SvnConexion j2EE module

Copy SVNConexion.war file in your tomcat deploy folder while its running, the j2EE module will be correctly set up after this action. The file has every library the module needs for working after the deploy. Remember here in wich folder you have deploy it for later fve configuration. (Normally this parameter is just the file name SVNConexion).

5. Configure FVE php module

For this purpose you have to access FVE with administration grants.

5.1 Access FVEProjectManager module.



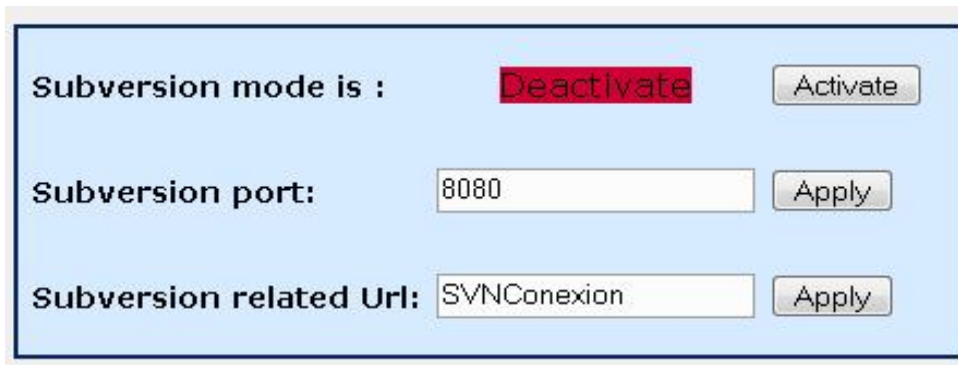
Follow the links to access configuration pane

5.2 Select Subversion link to access configuration pannel.



The FVE Subversion Menu access

5.3 Configure Subversion paramteres with the values established in your tomcat configuration.



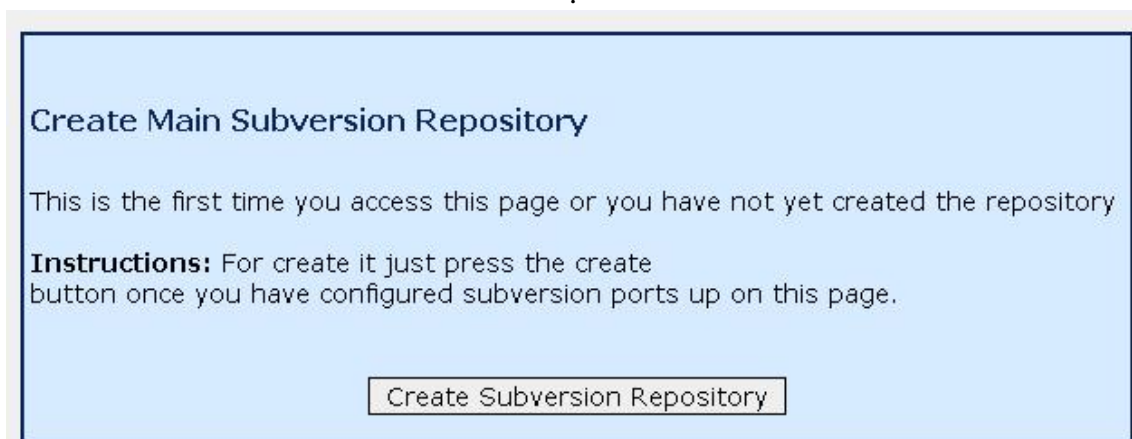
Subversion mode is : **Deactivate**

Subversion port:

Subversion related Url:

Subversion configuration Panel (with default configuration)

5.4 Press to activate the module. Window will refresh with a new panel option for create main repository under the previous panel.



Create Main Subversion Repository

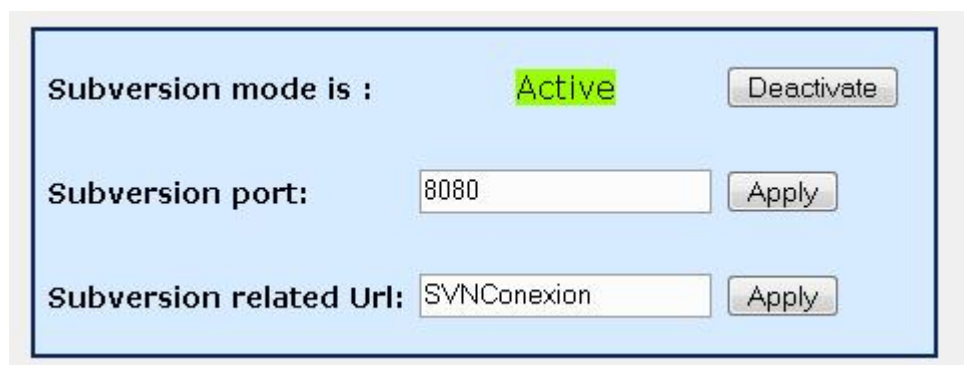
This is the first time you access this page or you have not yet created the repository

Instructions: For create it just press the create button once you have configured subversion ports up on this page.

Initial repository creation panel

5.5 Press the create repository button to generate it.

5.6 If all the steps were made correctly the panel option will disappear and you are able to work with FVE subversion.



Subversion mode is : **Active**

Subversion port:

Subversion related Url:

Congratulations, you have activated your Subversion module correctly!

Configure a subversion server "svnserve"

At this point subversion module works on localhost and can just be accessed on web. Subversion server "svnserve" allow us to access it on a console (or applications like tortoise) using a svn:// url path. For configuring this server please read subversion info in subversion.tigris.org.

If you don't have that much time pressing svnserve --help on you console may help you to configure it.

The easiest way to generate it under both windows and unix is to create a daemon pressing this command:

```
>svnserve -d -r "Local path of the repository"
```

Comando svnserve.

The local path of the repository is:

"dotproject_path/modules/fve/Subversion/Repositorio".

F.4. Linux installation

If you're using Linux, the installation process is quite similar to windows installation (described above) but there are some issues you have to have in mind:

FVE module needs uploading of files. So, you have to change the permissions of a couple of folders in order to allow apache write on those directories. These directories are the following (let's assume that you installed dotProject with FVE in the most common directory: www): **/var/www/dotProject/modules/fve/uploadedTemplateFiles** and **/var/www/dotProject/modules/fve/uploadedXmlFiles**. Ensure that you set enough privileges to write in these directories from your apache web server.

If you aren't able to see **Gantt Charts** in your dotProject with FVE application, the most probably reason is that you don't have the GD Library correctly configured. If you need to install it, follow the next steps:

- Install the GD Library for your PHP version. If you have apt-get service installed, try the next command in your linux console: apt-get install php5-gd.
- When you've installed the GD library, activate it on your php config file (php.ini) by adding the next line at the end of the php.ini file: extension=gd.so

That should be enough to make your Gantt Charts appear.

Thanks for installing dotProject with FVE Project Manager!



FVE logo.

Anexo G– Manual de usuario

El siguiente manual ha sido obtenido del manual de usuario que se incluye en formato HTML dentro la aplicación.

Es posible que no se visualiza de igual forma que en el navegador web debido a la adaptación desde HTML.

La finalidad de este manual es orientar a los usuarios para que puedan trabajar con la aplicación FVE Project Manager.

G.1. Introduction

What's FVE Project Manager?

FVE Project Manager is a modified version of dotProject which extends its functionalities. It was firstly created to work as a module for dotProject but, the project increased its size fastly, so more features were needed. The integration with dotProject core was necessary in order to provide an easy and simple way to work with it. In order to help users to work easily in their projects, FVE includes **Subversion** features to store projects source code.

Why use FVE Project Manager?

There are many new features that you can find inside FVE. The FVE project Manager was intended to include features that could be found in the program EPF composer. **Eclipse Process Framework** (EPF) is an open source project that is managed by the Eclipse Foundation. It lies under the top-level Eclipse Technology Project. EPF Composer is an open source process management tool platform and conceptual framework, which provides an easy to learn user-experience and simple to use features for authoring, tailoring, and deploying of development process frameworks.

When you are working with EPF Composer, you can create your own EPF processes adapted to your needs. Once you have created your custom process, you are able to export it as an XML file. This XML file contains all the information of the EPF process, including tasks, descriptions, roles ... etc. It is like a project mold. What FVE Project Manager does is taking this mold and fill it with all the necessary items which are required to begin working in that project.

What are these new functionalities of FVE Project Manager?

DotProject is a free project manager program based on web browser. DotProject is used as a foundation for FVE Project Manager. With dotProject you can:

- Create projects with tasks
- Create users and relate them with project and tasks.
- User admin, with logging permissions.
- Etc.

All these features are preserved in FVE, but as we use EPF Composer information, we were forced to include new features to be able to manage all EPF info. The new features that can be found in FVE are:

- Projects and tasks extended information.
- Import of XML files created with EPF Composer.
- Workproduct management.
- Templates linked to workproducts.
- Extended role information, which make users capable or not to work in a task or a workproduct.
- Subversion features.
- And so on.

The standard structure of dotProject is respected, and both the people who had previously worked with dotproject as new users will feel comfortable working with FVE.

How can I use this user manual?

This manual is divided into several paragraphs, each one of them will help you in one aspect of the application. The manual paragraphs are the following:

Introduction: the current section. It gives general information about the application.

Installation instructions: if you have to do a deployment of the application, this paragraph will help you through the process.

Creating a new project: when you want to start a new project, you have two main ways of doing it. Follow the instructions inside this section to learn how to do it.

Modifying your project: once you have your project created, you will probably want to make changes on it, such as changing users or creating new elements. This section will help you to do it in a simple way.

Subversion integration: if you want to add a SVN repository to your project, this section will tell you how can you do it.

Tutorials: perhaps it simpler following an example step by step to learn how to do some action in the application. In this paragraph, you will find several tutorials that will help you understanding what are the steps that you have to follow to complete an objective.

Glosary: all the terms that are used in this manual and in the application are explained in detail in this section.

We hope that this manual will be helpful to you for handle through our application.

Thank you very much,

FVE Team.

G.2. Creating a new Project

Creating a project

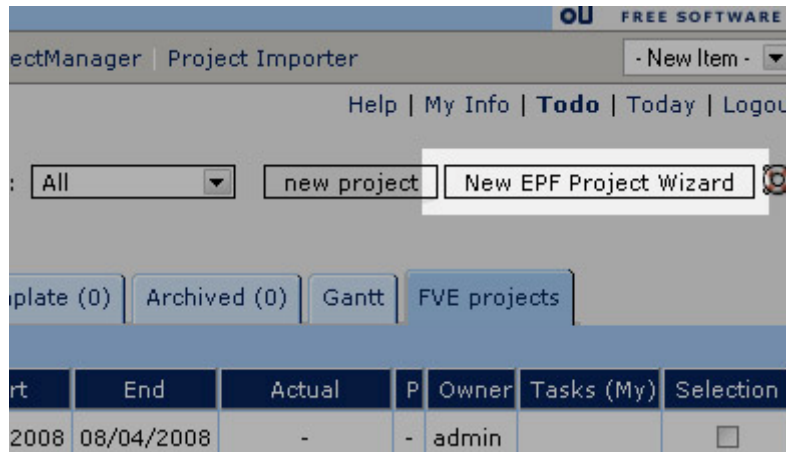
If you want to work with dotProject or with FVE Project Manager, you have to create project. In general terms, a project consists on a group of tasks where the users will work. Each task has objectives and resources associated to it.

There are two different ways to create a project: manually or using FVE's Project Creation Wizard. After these lines, both methods are explained in simple terms.

Project creation using FVE's Project Creation Wizard.

A wizard is an assistant who helps users to do some specific task. FVE implements a Project Creation Wizard which helps users creating new projects importing the project mold from an XML File created with EPF Composer. This wizard consists of eight steps, each of them make a different function to configurate and create the new project. After those eight steps, the project will be created and the users will be able to work in ther assigned tasks.

To launch this wizard, you can use the "New EPF Project Wizard" located in the Projects module.

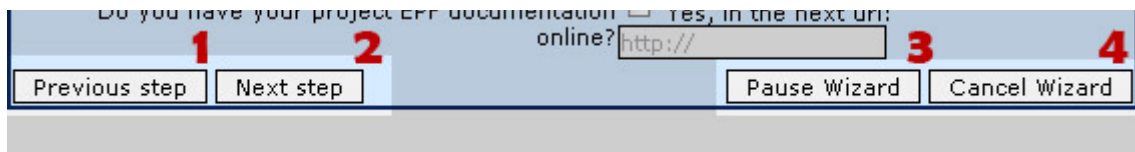


The button that launches the Project Creation Wizard

Once you've pressed this button, you will be in the Project Creation Wizard page. This wizard consists in eight steps. We will explain in detail each one of them, but before that it's important to know how does the wizard works.

Wizard control buttons

In each step (except the first and last) you can find four buttons at the bottom of the wizard form.



These buttons control the flow of the wizard steps

As you can see in the image above, these buttons allow you to move through the wizard. It's also possible pause the wizard and come back to it in future keeping all the changes you already did. Those four buttons have the following functions:

- **Previous step** button: you can go back to the last step of the wizard. The changes you made in the current step will be saved.
- **Next step** button: you move forward to the next step of the wizard. The changes you made in the current step will be saved.
- **Pause Wizard** button: you pause the creation of your project. The changes will be retained until you launch again the project creation wizard. You will return at the same step you pressed the pause button.
- **Cancel Wizard** button: it cancels the current project creation. All the changed you made will be discarded.

Once you know how to move through the wizard, let's explain in detail what does each step do. Let's begin with the first one: project creation.

1st Step: Project creation

This is the first operation you have to do when you create a project: giving it a name and its main data.

First step form: filling project main data

You have six fields that you have to fill in to create your project:

- **Project name:** the name of your project. It will represent it through all his life, so it's highly recommended you to write a meaningful name.
- **Company:** this is the company that will hold the project. You can choose among all the companies you have already created in you application.
- **Owner:** the person who will be responsible of the project. Choose among all your application users.
- **Start and finish dates:** these are the dates that you stimate your project will accomplish. To introduce them, press the calendar button (📅) and a pop up window will open with a calendar on it. Choose the day you want and the field will be filled with that date.
- **Description:** a short description that explains what does your project consist of.

Once you have filled these fields, we can go on to the next step: project import.

2nd Step: Project import

This is perhaps one of the most important step in your project creatin. You have the chance of importing an XML File created with EPF Composer that contains all your project data. You will also create a repository for your new project.

The screenshot shows a wizard window with a light blue background. At the top, it asks 'Do you wish to import from an EPF XML file?:' with a checked checkbox. Below this, there's a text field containing 'c:\xmlunicoPlugin_xp.xml' and a button labeled 'Examinar...'. The next section asks 'Do you want to create a new repository or use an existing one:' with three radio button options: 'Create a new one', 'Don't use a repository', and 'Use this repository'. The 'Use this repository' option is selected, and there's a text field next to it containing '192.168.2.45'. The final section asks 'Do you have your project EPF documentation online?' with a checked checkbox and the text 'Yes, in the next url:'. There's a text field next to it containing 'http://intra.dot/info.html'. At the bottom, there are four buttons: 'Previous step', 'Next step', 'Pause Wizard', and 'Cancel Wizard'.

Second step form: importing from an XML File and creating your repository

There are 3 main parts in this form, the first one is to upload an XML file from your file system. The second one is to choose what kind of repository do you want to use. And the third one is the choose an online route where you have your EPF documentation uploaded. This can be very useful if you wish to query any data refered to your EPF project structure. So, to complete correctly this step, you have to fill in the next fields.

XML File: if you want to upload an XML File (it's optional, anyway) check the box and then press the "examine" button to find the XML file you want to import from your file system.

Repository: choose the action you want to do with the repository: you can create a new one in your current server; you can disable the creation of the repository or you can use an existing repository giving it's URL. This last option is still under development, so it won't probably work as you could expect.

EPF Guidelines: this field contains the direction where you have uploaded the documentation of your EPF project. You can query it whenever you want during the project life. If you want to use this feature, check the "Yes" box, and the write the URL of you EPF Guideline documentation.

Be careful, because once you have imported a project from an XML file, you won't be able to come back to this step any longer. This is because the structure of your project is defined at the moment you import the XML. Changing all this structure could cause problems to your project. So, if you want to import from another XML file, just cancel your current project and make a fresh start on this wizard.

When you've finished, you can jump to next step and begin configuring your project users.

3rd Step: Project users

Obviously, you want that there are users working on your project. To add them, you have to do it at this point.

Third step: assigning users to your project

This operation is very simple, you only have to select one user from the "Available users" list and then press the "Add user to project" button. You need to have the users created to appear in the available users list. Below these fields, you will find a list where you can see all the users that you have inserted into your project. If you want to remove one of them, just select it by clicking on it and press the "Remove user from project" button. The user you selected will go back to the available user list.

Now that you have selected the users that you want to work in your project, let's advance to the next step and configure the users roles.

4th Step: Configuring roles

At this point, you have your users inserted in your project. Now you have to decide what roles are they qualified to play. Note that this decision is very important because in further steps (in workproducts and tasks steps) you will associate users to charges according to its role.

Fourth step: assigning users to available roles

In the previous image you can see a single role form. There will be as much of these forms as roles are imported from the XML file imported. In each of these roles you will have to assign the users that you think can play the role

indicated in the "Role" row. Some of these roles can have a description linked to them that can explain in detail the proposal of the role. To add a user to this role, just select it from the available users list (these users are those included in previous step) and press the "Add user to role" button. You will notice that the selected user will appear in the assigned users list. If you want to remove a user from this list, just click on it and press the "Remove user from role" button. The user will come back to the available users list.

At this point, your project users are already set up. Now we're going to the next step of the wizard: configuring workproducts.

5th Step: Configuring workproducts

This is one of the new features included in FVE project Manager. Workproducts are elements that can be inputs or outputs of tasks. They could have a template file associated to them. The workproducts can be related to one or more tasks.

Fifth step: configuring a workproduct

There are several things that you have to set up in a workproduct to correctly complete this stage of the wizard. As you can see in the previous image, these sections are marked with numbers from 1 to 3. Now, let's explain each of them:

1. **Workproduct responsible:** this person is the one in charge of the workproduct. It has to play the role showed by the role in brackets. Choose him/her from the drop-down list. One a unique user can be the workproduct responsible.

2. **Workproduct modifiers:** these people will be able to modify the workproduct status. A workproduct can have many roles modifiers as you want (in this case, as have been imported from the XML file). For each workproduct modifier role, there can be zero or more users, it depends on how many users play the specific role. This role is pointed out next to the "Role" row. Add and remove users using the "Add user" and "Remove user" buttons.

3. **Workproduct templates:** this is an optional field for the workproduct. You have the option of adding a template file the workproduct. Only a single template file is allowed. If you want to add a new template, the older template will be deleted. If you want to add a template file, check the "Yes" box, and select it from your file system by pressing the "Examine" button. Now, fill in the "Name" and "Description" fields. If you go to a different step and then you come back again to this workproduct step, if you previously created a template file, you will see something very similar to the following image:

Template name	DocFile
Template description	Contains all the information
Template path:	./modules/fve/uploadedTemplateFiles/02008-05-20-210945.txt

☒ Yes (change): Name: Description:

A workproduct with a template associated to it

Here you can see that you have a Template file associated to the workproduct. You can do several things:

- Leave it as is: Just don't touch anything and the workproduct will remain its template file.
- Delete it: to do so, just uncheck the "Yes" box, and when you go to the next/previous step, the template will be deleted from the workproduct.
- Modify it: if you want to upload a different file, press the "Examine" button and select a different file from your file system. If you only want to modify the template name or description, change it directly from the "Name" and "Description" fields.

Be careful because if you import an XML File that has template files associated to workproducts, you can see something very similar to the previous image with the difference that there is no uploaded file for the template. So, if you want to fix this, just upload again the file you want to be the template file.

Now that you have configured your workproducts let's go to the next and very important step: configuring tasks.

6th Step: Configuring tasks

Tasks are the core of your project. In these tasks, there will be workers that will complete them and the workproducts that are part of them. The configuration of the tasks is very simple using FVE's wizard.

27 Task: **write_user_story**
Description:

Configuring Users
 Primary: Camunias, Pepito (xp_customer) **1**

Additional
 Role: xp_tester
 Available users: ▼ Add user
Pérez, Pedro
 Modifier users: ▼ Remove user
 Role: xp_programmer
 Available users: ▼ Add user
Person, Admin
 Modifier users: ▼ Remove user **2**

Adjusting Dates
 Start date: 22/04/2008 10 : 00
 Finish date: 27/04/2008 11 : 00
 Duration: hours Calculate Duration **3**

Linked Workproducts

Name	Description	Dependency Type
xp_vision	Defines the stakeholder's view of the product to be developed, specified in terms of the stakeholder's key needs and features.	optionalInput
xp_user_story	A brief description of some functionality provided by the system from the point of view of a user of that system.	output

4

Sixth step: configuring a task

For each task the project imported, you will see a form very similar to the upper image. As you can see, there are four different parts in this form, that let's explain below:

- Task primary user:** this is the responsible of the task. It's a single user that has to play the correct role (which is indicated in brackets). You can select him/her from the "Primary" drop-down list. These are the users that can play the correct task primary role.
- Additional users:** as happened in the workproducts modifiers, you can add as much users as you want as long as they fulfil the correct modifier

role (indicated next to "Rol" row). Add and remove user from the modifiers lists using the "Add user" and "Remove user" buttons.

3. **Task dates:** you can set the task dates with this controllers. You have to configure a starting date (with day and hour) and a finish date (with day and hour too). To do this, select the day pressing the calendar button (📅). A pop up window will appear. Select the day in this window by clicking on it. Once you have selected the day, choose the hour deploying the hour drop down list. Do the same process to the minute drop down list. Once you have configured both starting and finish dates, you can check how much hours or day this task has. Select hours or days in the duration selector and press the "Calculate Duration" button to see the duration of the task in the neighbor box. This functionality has no impact on the task parameters.

4. **Linked workproducts:** in this table you can see all the workproducts that are related to this task. You cannot modify them from this point. You will have to do it manually in the dotProject interface later.

7th Step: Finishing your project

This is the last step in the creation of your project. This step do nothing, but you have the chance to come back to the previous steps in order to modify any paramater you want. Just press the "Previous step" and head to the step you want to modify. Once you're pleased with your project, press the "Finish" button and your project will be created.

Now, you have just successfully created a new project into your application. You can start working on it with your team. If you want to edit the parametres of your project once it has been defined, please go to the "Modifying your project" section of this manual.

If you want to create a project manually, just keep reading the next paragraph of this section.

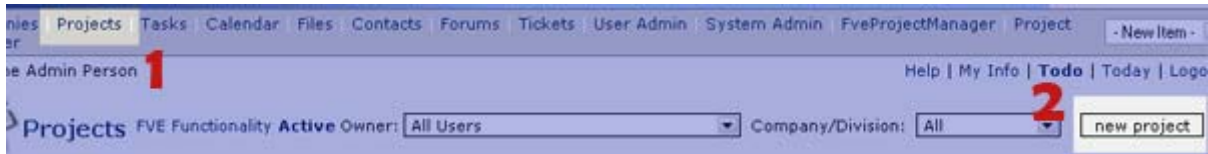
Creating your project manually.

If you don't want to import your project from an XML file but you want to include the new functionalities that FVE Project Manager offers, you can create your project manually following the instructions that are included in this paragraph.

First of all, you have to know that when you create a new project in dotProject, this project won't have any FVE functionalities on it. The same will happen with the tasks. To switch them to FVE project and tasks, you have to make a very simple conversion. Bellow, you will be teached how to do it. The instructions contained in this paragraph won't be as detailed as in the previous one, because many of them are explained in the dotProject user manual ([link](#)).

Creating your project

To create a new project, just go to the Projects module and press the "New Project" button.



Creating a new project

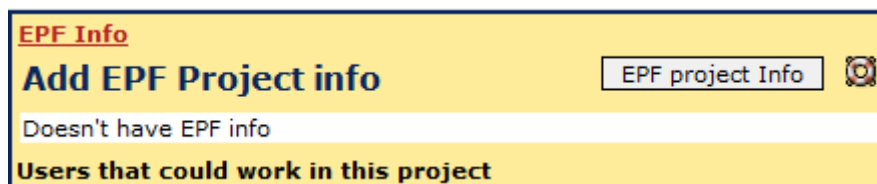
Once you've done this, a new creating form will appear in your screen. Fill this form according to your needs, and when you're ready, press the finish button.

When you have created your project, it will appear in the Project Module > All Projects tab. As you can see, there is a FVE Projects tabs. If you had created your project with EPF Project Creation Wizard, it will appear in this tab, but as we created the project using the common way, it doesn't appear yet in this tab.

Converting your project to an FVE Project

What we have to do now is converting your project into a FVE project. This is a very simple process that can be accomplished in very few steps:

Firstly, go to your new project info screen. To do so, just press on your project name in the Projects Module > All Projects Tab. If you've done this correctly, you will see a screen very similar to the next one:



Viewing the project info

As you can see, here you can find all the info of the project you created. You can notice that the remarked area of the image is the EPF info of your project, and it doesn't exist yet. To convert your project to a FVE Project, you have to add to it EPF info. To do this, just press the "EPF project info" button.

When you've pressed this button, you will jump into a EPF Info form. It will look like this:

Converting your project into a EPF project

As you can see, there's a couple of fields where your project name and description can be found. In this form, you have to fill the following EPF information:

- **Guidelink Link:** This is the URL where you have your Project process information uploaded. It will be able to be consulted by your users to obtain information about a specific item of your project like a task or a workproduct.
- **Respository path:** If you want to use a SVN (Subversion) repository to support your project development, just fill the field with the path of your repository.

Once you've filled these fields, press the "Edit Project" button. Now you will go to the next screen, where you have to add the users that you want that work in your current project.

Converting your project into a EPF project: Adding users

This form is very simple and you only have to choose the users from the "Available Users" drop down list and press the "Add user" button. If you want to remove a user from the "Assigned Users" list, just press the "Remove User" button. When you're done, just press the "Finish EPF Editing" button.

Now, your project has been converted into a FVE Project. You can see it both in the tab "All Projects" and "FVE Projects". If you go now to your project info screen (by clicking on its name), you will notice that the new EPF info will be showed in a new project info form.

EPF Info	
Edit EPF Project info	
Subversion repository path:	- none -
Guideline link (EPF):	Doesn't have
Plugin used (EPF):	SimTrafficProcess
Plugin description (EPF):	Main plugin
Plugin authors:	Ignacio Estirado, Íñigo Fábregas, Alejandro Vilela, with the experience of other 17 workers in SimTraffic project.
Users that could work in this project	
User 1	admin
User 2	Paco
User 3	analyst
User 4	project manager
User 5	stakeholder
User 6	architect
User 7	berto

The new EPF info of your project

You can always modify this info pressing the "EPF project info" button.

You have created your project. Now you'd probable like to add tasks to it. Read the next paragraph to learn how to do it.

Creating tasks

To create a task inside your project, you have to go to your project main view (pressing its name in any part of the application where it appears). Then, in the upper zone of this info screen, you will see several buttons to create multiple items on your project. Press the button where it says "New task".



The button to create a new task in your project

After pressing this button, you will be moved into the New Task form. Please, fill in the fields according to your preferences, and when you've finished press the save button.

The new task will be created in your project, but it's still a current dotProject task, to learn how to give it FVE functionalities, read the next paragraph.

Converting your tasks to an FVE Task

In order to use the new FVE functionalities in your tasks, you have to make a conversion in them. To do this, go to your project info page, and open its

tasks tab. Look for the new tasks you've just created and click on its name. The tasks info screen will show up, and it will look very similar to this:

The EPF Info of your task is empty

As you can see, there is no FVE Task info associated to your tasks, so if you want to add it, press the "EPF task Info" button. After doing this, you will enter into a small wizard where you will go through several step in order to configure the EPF info of your task. It's highly recommended that you've already configured your project roles, users and workproducts because if you have done that, you will be able to stablish the relationship among all of them. To learn how to create an configure your project items, please go to the "Modifying your project" section.

Ok, let's assume that you've configured your project users and roles and you've created several workproducts. When you press "EPF task Info" button, you will enter into a small wizard where you will configure your taks additional FVE info. The first screen of this wizard is the next one:

First step in your FVE task configuration

This first step is very simple: you only have to set your task type. Set the type you think best fits your task, and when you're done, press "Next step" button. You will jump to the next step in the task configuration.

Second step in your FVE task configuration

At this form, you have to configure both tasks role information and task workproduct data. To do this, you have to follow the next instructions:

1. **Role's information:** you have to select what roles you want to work in your task. You have to select a single primary role and zero or more additional roles. To do this, just select the role from the drop-down list and press "Add Additional Role". If you want to remove a role from the assigned roles for Additional, just press the "Remove additional role" button after selecting the desired role.

2. **Workproduct's information:** here you can select what workproducts you want to have bearing with your tasks. Choose your workproduct from the Workproduct list and then choose the type of dependency you want to establish between your task and the workproduct. Once you've chose this, press the "AddTaskDependency" button. If you want to remove one inserted dependency, click the workproduct name in the inserted workproduct list (the list on the left) and after that, press the "Remove task dependency" button. The inserted workproduct dependency will disappear from the list.

So when you're configured this, press the "Next step" button to head to the next and last step in your tasks configuration.

The third and last step in your FVE task configuration

The previous image shows the last step in your FVE task configuration process. In this form, you have to choose the users you want to associate to your task. These users are playing the roles you established in the previous step. You have to choose a single user for the primary position (1) and for each modifier role (2), you can add as many users as you want. To do so, just press the "Add user" button when you've selected the user in the available user drop down list. If you want to remove a user, click on it and press the remove user button.

When you are satisfied with the changes you made in your FVE tasks, press the "Finish" button and all your changes will be saved.

If you want to know further information about how to modify your project, please refer to next section of this manual: Modifying your Project.

G.3. Modifying your Project

You have a FVE Project. And now, what?

If you want to work with dotProject or with FVE Project Manager, you have to create project. In general terms, a project consists on a group of tasks where the users will work. Each task has objectives and resources associated to it.

EPF Project Importer is not enough. I need to create some others entities manually.

You will probably ask yourself this question. Most users need to extend the stuff imported from the XML project because the life of a project is very dynamic and you will probably need creating extra tasks, workproducts and templates inside your project. In this section you will find how to create this elements and how to work with them, showing or editing them. You will be able to work with the next elements:

- Tasks
- EPF Roles
- Workproducts
- Templates

Don't be afraid of using them inside your project, the tools brought by our system to you are very simple and intuitive so, try all the functions!

Creating your roles manually

If you want to assign responsibilities to some users for specific functions you probably want to create a role. This structure is also called EPF Role because it has the same structure that is created in the EPF importer module for roles. If you want to create a new role in a project, you have at least two options.

First possibility:

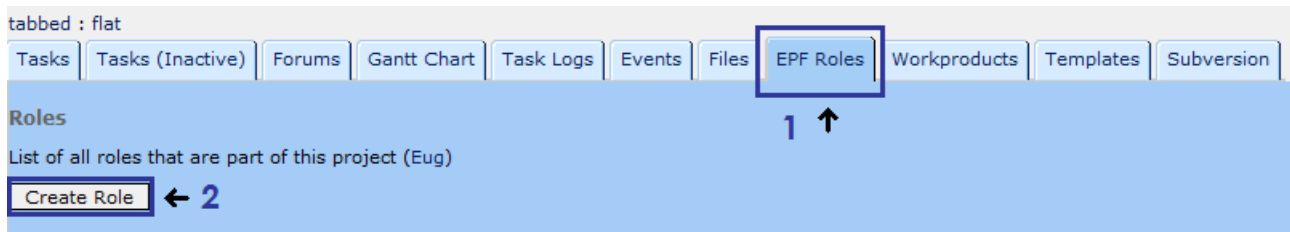
- 1 - Enter in the Project module.
- 2 - Enter in the concrete project in which you want to create the new role, in section View Project.
- 3 - Press the New Role button, located in the upper Tab that establish the title of module Projects.



The button that launches the Role Creation Wizard in the button "New role"

Second possibility:

- 1 - Enter in the Project module.
- 2 - Enter in the concrete project in which you want to create the new role, in section View Project.
- 3 - Press the Tab called "EPF Roles", that is located in the Tab group that is down to project view.
- 4 - Press the button called "Create Role" in the new section that Roles Tab had opened.



The button that launches the Role Creation Wizard in the tab "EPF Roles"

When you enter in the Create Role section, you start a little Wizard in which you can specify role attributes and relations with other entities. The first form that you can see when you enter in this section is like this image:

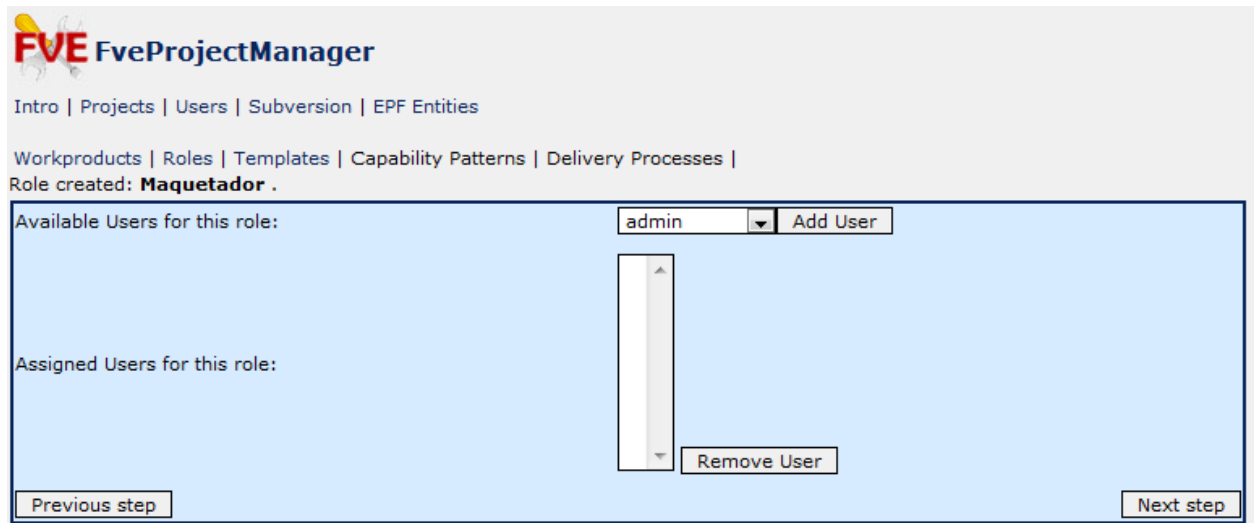
The first step in EPF Role Creation Wizard.

In this first step you can specify General Information about this role, as his name, his description and project that it belongs to. To fill the project attribute you should select the chosen project in a combobox of every FVE Projects in the application. If your project doesn't appear, probably this project isn't yet an EPF Project and is only a dotProject Project.

When you have filled the form, you can push the "Next step" button. If name or description of the new role is empty, the application alert you with a pop-up with this message: *"Please, check the value of attribute role_name (actually null)"* and force you to enter values on this attributes.

If everything goes well, the browser should change the web and redirects you to the next step of this wizard. This step has the function of assign the users that can assume the role in the project that you previously specified. You can select only the users that you previously inserted as

workers of this project, and not all users. The aspect of this step will be like this image:



FVE FveProjectManager

Intro | Projects | Users | Subversion | EPF Entities

Workproducts | Roles | Templates | Capability Patterns | Delivery Processes |

Role created: **Maquetador**.

Available Users for this role: admin Add User

Assigned Users for this role:

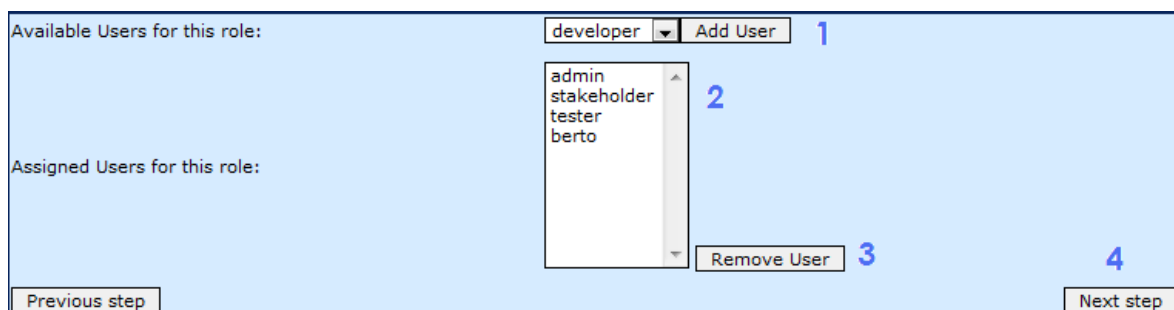
Remove User

Previous step Next step

The second step in EPF Role Creation Wizard. Initial aspect.

As you can notice, the first thing that appears in this step is a message that informs you that the Role was created (with the name of the role, in this case "Maquetador"). In the form you can also see the list of workers of the project (Available Users for this role) in a combobox, and a listbox (Assigned Users for this role) initially empty. There are two buttons, "Add User" and "Remove user", that let you add and remove the users of the project. Concretely, the "Add User" option adds to the list box the selected user in the available checkbox, and removes it from available users. If you had made a mistake and had added a user that really can't assume this role, you only have to select the user in assigned users and push the button "Remove User". Two operations are implemented in Javascript, so you don't leave this form until you press next button.

When you selected the users that you want, you probably have a list like this:



Available Users for this role: developer Add User 1

Assigned Users for this role: admin stakeholder tester berto 2

Remove User 3

Previous step Next step 4

The second step in EPF Role Creation Wizard, with Users selected

Then you should press button "Next Step", and will be redirected to a new window that confirms the creation of the role with specified properties. Besides it, the application suggests you the option of show the role that you have created previously. At this moment you have a new role and a list of users with it in your project, prepared to enter in new tasks and workproducts.



The last step in EPF Role Creation Wizard, alert of role created

Creating your tasks manually

Create a new tasks is a very simple process that you can make in a few minutes. First, you need to choose the project where you want to create you new tasks. Once you have decided what project you will use, go to its main information page.

As you will see in the top of the screen, there are several options you can carry out: create a new task, create a new role or a new workproduct. Let's focus on the "New task" option. Just press that button to enter in the task creation page.

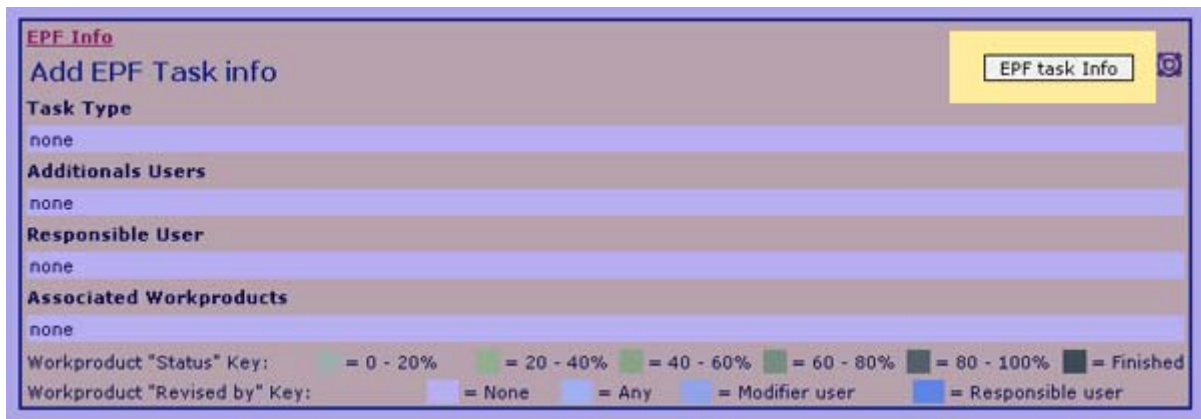


Press this button to create a task in your project

The task creation page is very intuitive and it allows you setting the characteristics of your new tasks. There are a lot of fields that you can fill, but the most important are:

- Tasks dates: when does it start and when does it finish.
- Task name and description: they will help you identifying your task in future.
- Task parent and dependencies: does your task depend on another one?

Once you have created your task, it will work as a usual dotProject task. If you want to add EPF information to it, you have to do a very simple thing: in the center of the task info screen, you will see the EPF info table. Now, there is no info added to your task. To do so, press the button you can see in the next image:



Press this button to add EPF info to your task

Once you've pressed this button, you will enter in the tasks EPF info wizard. It's very simple and it only consists in three steps:

- 1st. Configure your task type: enter the text you think is more suitable for you task.
- 2nd. Role and workproduct information: select the roles that the users of your system must play in order to work in your tasks. Also configure the relationships of you task with the workproducts of the project it's inserted.
- 3rd. Choose the users that will work in the task: select the users according to step 2 that will work in your task.

Now you have just created a new task into your project and you inserted EPF info to it!

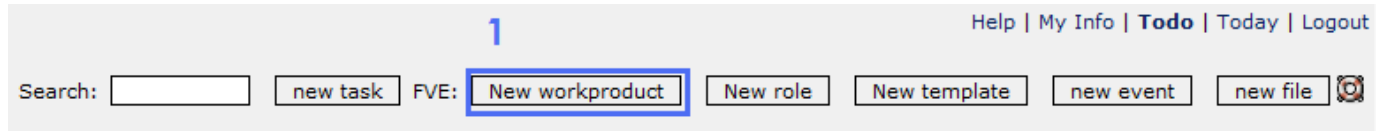
Creating your workproducts manually

If you want to define a document or other type of artifact that is used, produced or modified in the project and/or in some tasks, you probably want to create a Workproduct. This structure is defined as EPF Composer defines workproducts, and therefore a workproduct has a responsible user and could have some users as modifiers. If you want to create a new workproduct in a project, you have at least two options.

First possibility:

- 1 - Enter in the Project module.

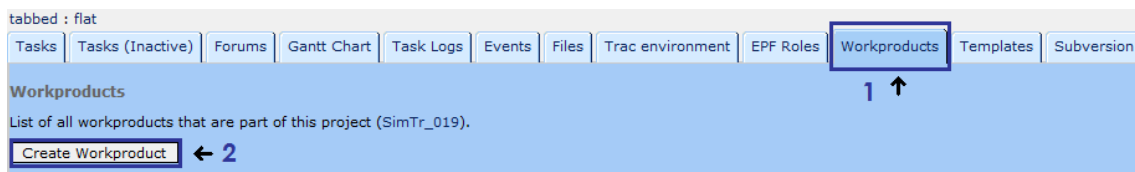
- 2 - Enter in the concrete project in which you want to create the New workproduct, in section View Project.
- 3 - Press the New Workproduct button, located in the upper Tab that establish the title of module Projects.



The button that launches the Role Creation Wizard in the button "New role"

Second possibility:

- 1 - Enter in the Project module.
- 2 - Enter in the concrete project in which you want to create the new workproduct, in section View Project.
- 3 - Press the Tab called "Workproducts", that is located in the Tab group that is down to project view.
- 4 - Press the button called "Create workproduct" in the new section that Workproducts Tab has opened.



The button that launches the Role Creation Wizard in the tab "EPF Roles"

When you enter in the Create Workproduct section, you start a little Wizard in which you can specify worproduct attributes and relations with other entities (tasks and roles). The first form that you can see when you enter in this section is like this:

General information:	
Name:	<input type="text"/>
Brief description:	<input type="text"/>
Project:	<input type="text" value="www (id:13)"/>
<input type="button" value="Back"/>	<input type="button" value="Next step"/>

The first step in Workproduct Creation Wizard.

In this first step you can specify General Information about this workproduct, as his name, his description and project that it belongs to. To fill the project attribute you should select the chosen project in a combobox

of every FVE Projects in the application. If your project doesn't appear, probably this project isn't now a EPF Project and is only a dotProject Project.

When you have filled the form, you can push the "Next step" button. If name or description of the new workproduct is empty, the application alerts you with a pop-up with this message: *"Please, check the value of attribute workproduct_name (actually null)"* or *"Please, check the value of attribute workproduct_description (actually null)"* and force you to enter values on this attributes.

If everything goes well, the browser should change the web and redirects you to the next step of this wizard. After this you should know that the Workproduct has been created. This second step has the function of assign the task dependencies of this workproduct in the project that you previously especificated. You can select only the tasks that belong to this project, and 3 defined types: Optional input, Mandatory Input and Output. The aspect of this step will be like this:

The second step in Workproduct Creation Wizard. Initial aspect.

In this form you can see 2 sections. The options in the upper section are referred to the role dependencies: You can select the role responsible of the workproduct (only one) in a combobox that has all roles defined in this project. Also you can select additional roles that act as modifiers of this workproduct in a listbox initially empty. There are two buttons, "Add Additional Role" and "Remove Additional Role" that let you add and remove the additional roles in this workproduct. Concretely, the "Add Additional role" option adds to the list box the selected user in the available checkbox of, and removes it from available roles. If you had made a mistake and had

added an additional role that really can't modify this work product, you only have to select the role in "Assigned Roles for Additional" and push the button "Remove Additional roles". Two operations are implemented in Javascript, so you don't leave this form until you press next button and you can see the actual relations.

The options in the lower section are referred to the tasks dependencies: You can select the task that has relation with this workproduct in a combobox that has all tasks defined in this project. Also you can select the type of the dependency in a combobox with 3 options: Mandatory input, Optional input and Output. When you selected both, task and type dependency, you can push the button Add Task Dependency, and this info will be added to the two listbox that is located down. If you had made a mistake and had added a task dependency that really doesn't exist, you only have to select the task in the left listbox and push the button "Remove task dependency". Two operations are implemented in Javascript, so you don't leave this form until you press next button and you can see the actual relations. In this case, when you add a task dependency the task listbox, the name of the task doesn't disappear of the task checkbox, because is possible to add the same task more than one time with different type dependency. Only if you want to add a task dependency that you have previously inserted, the application would show you a alert window with this message: *"Sorry, you can't introduce a redundant task dependency (you had added it previously)"*. Due to this alerts, when you click in the "Next step" button you are sure that the task dependencies are introduced correctly.

When you selected the roles and tasks dependencies that you want, you probably have a list like this:

Editing an existing WorkProduct

In this page you will be able to edit one workproduct who belongs to a task of one of your projects. Please follow all the steps in order to correctly.

Role responsible:	xp_system_administrator	1
Available Roles for Additional:	xp_customer_req	2
Assigned Roles for Additional:	xp_implementer xp_tester xp_designer xp_programmer	3
Available Tasks for Dependencies with Workproducts:	Task-> update_release_plan Output <-Type	4 5
Assigned Dependencies between Tasks and Workproducts:	AddTaskDependency estimate_task develop_xp_vision develop_xp_vision define_customer_test update_iteration_plan update_iteration_plan update_release_plan update_release_plan OptionalInput OptionalInput Output Output MandatoryInput Output MandatoryInput Output	6
	Remove task dependency	7
		8

Back Next step

The second step in Workproduct Creation Wizard, with roles and tasks selected

Then you should press button "Next Step", and will be redirected to a new window that reads the values of roles responsible and modifiers introduced and shows you a new form that let you choose the users responsible and modifiers of this workproduct. You can see a list of users in every role selected: This list contains every user that participate in the project and can assume this role. You can select one responsible user, and one or more modifiers users in every modifier role. Also is possible don't introduce users for all roles, if you consider it better.

In order to add user responsible with previously selected responsible role, you should select him for the combobox responsible (1). To add users as modifier users you have two buttons for every modifier role: Add user (2) and Remove user (3). You can select a user from a list of users that can assume selected role and added him with Add User button, and remove it if you selected the user in listbox of users and click in Remove User. When you have finished you have a form filled like this:

Configuring workproduct 564

Configuring Users

Responsible: (xp_system_administrator) 1

Modifiers

Role: xp_implementer

Available users: Add user 2

Modifier users: Remove user 3

Role: xp_designer

Available users: Add user

Modifier users: Remove user

Role: xp_tester

Available users: Add user

Modifier users: Remove user

Role: xp_programmer

Available users: Add user

Modifier users: Remove user 4

[Next step](#)

The third step in Workproduct Creation Wizard, Configuring users for roles selected

Then you should press button "Next Step", and will be redirected to a new window that confirms the creation of the workproduct with specified properties. Besides it, the application suggests you the option of show the workproduct that you have created previously.

Editing an existing WorkProduct

Success!

✓ Workproduct correctly created: Plan Project , with specified properties.

[< Go back to workproduct info page](#)

The last step in Workproduct Creation Wizard, alert of workproduct created

Creating your templates manually

Maybe you would like to create new template into your workproducts. First you need to know that you can only create one template for each workproduct. And you also need to upload a file to this new template.

Ok, now that you know the rules, let's create a template. In the info page of the project you want to create your template, press the button "new template" in the upper zone of the page:



The button that launches the wizard to create a new template

You will access to the template creation wizard. This wizard only consists in two easy steps where you have to do the following actions:

- 1st Step. Choose a project for you template: choose from the dropdown list the project where you want to create your template.
- 2nd Step. Template information: choose the workproduct where you want to create the template. If you choose a workproduct that already has another template you want be able to create your template. Once you've chosen the workproduct where you want to create your template, fill the name and description fields and choose a file from your system to stands for your template.


When your're done, press the finish button and if everything went right, the template would have been created in your project workproduct.

Everything changes: I need to change information.

In every project you work, there will be changes. FVE project manager is ready to allow those changes and all the elements in the system are able to be modified at any moment. In the next sections, you will learn how to modify the data associated to your roles, tasks, workproducts and templates. Using these tools, your project status will always be the latest possible.

Editing your roles

Editing your roles is very simple. To do it, you only have to follow the next steps.

First, locate the role you want to edit. The best way to do it is going to the project which contains the role you want to edit. In its info page, open the "EPF Roles" tab and look for your role. Press the button  and you will jump to the role edition page.

The image shows two side-by-side screenshots of a web application interface for editing a role.

Left Screenshot (General information):

- Title:** General information:
- Name:** Project Manager
- Brief description:** He will be the leader of the project and the responsible of organization
- Project (Not possible to change):** Project Development
- Buttons:** Back, Edit Role

Right Screenshot (Users):

- Title:** Available Users for this role:
- Available Users:** Paco, sor, admin
- Assigned Users for this role:**
- Buttons:** Add User, Remove User, Back, Save changes


Two steps to edit a role

There are two things you have to edit in a role. They are displayed in separated pages:

- In the first page you may edit you role main values, such as its name and brief description You won't be able to change your role project.
- In the second page, you will tune the users that can play this role. Select them from the drop-down list where all the system existing users are listed. Press "Save changes" when you're done.

As you can see, in a few steps you've edited the information of your role.

Editing your tasks

Now we will explain how editing your project tasks. The operation is very simple. First of all, you have to find the task you want to edit. Do it by going to your project main information page. Now, expand the tasks tab. All the tasks inserted in this project will be listed. Look for the task you want to edit and press the  button. You will access to the tasks editing page. Make the necessary changed and when you're done, press the "save" button to commit your changes.

If you want to edit your task EPF information, you have to press the "EPF task Info" button that you'll find in the EPF Info box of your task info page. If the task have no EPF data associated to it, you will see at the top of this box the following message: "Add EPF Task Info". However, if the task has already EPF Info you can check it by the "Edit EPF Taks info" text at the top of the EPF info box.

EPF Info

Edit EPF Task info

Task Type
Pues una tarea

Additional Users
none

Responsible User
Paco SpokesMan Hablador

Associated Workproducts

Name	TypeDependency	Status	Revised by
PlanProject	OptionalInput	60 %	admin
OnePage Idea	output	0 %	None

Workproduct "Status" Key:
■ = 0 - 20% ■ = 20 - 40% ■ = 40 - 60% ■ = 60 - 80% ■ = 80 - 100% ■ = Finished

Workproduct "Revised by" Key:
■ = None ■ = Any ■ = Modifier user ■ = Responsible user


Press this button to edit your task EPF data

You will jump to a small wizard where you have to tune up the next features of your task:

- 1st. Configure your task type: enter the text you think is more suitable for you task.
- 2nd. Role and workproduct information: select the roles that the users of your system must play in order to work in your tasks. Also configure the relationships of you task with the workproducts of the project it's inserted.
- 3rd. Choose the users that will work in the task: select the users according to step 2 that will work in your task.

When you've finished your editing, press the "Finish" button and the EPF info of your task will be recorded into the system.

Editing your workproducts

If you want to edit a workproduct existing in your system, firstly you need to find it in your project. Open the "Workproducts" tab in your project main info page to list all the workproducts that exist in your project. Look for the workproduct you want to edit and press the  button to edit it. You will jump to the workproduct info page. In this page, there are several links to edit your workproduct information.

General information:
 Id: 35
 Name: xp_vision
 Description: Defines the stakeholder's view of the product to be developed, specified in terms of the stakeholder's key needs and features.
 Part of project: Fall out
 Edit general information **1**

Template associated:
 Doesn't have Associated Template
 Create template information **2**

Role's relations:
 User responsible: stakeholder
 Role responsible: xp_customer_req
 Doesn't have Modifiers Users And Modifiers Roles
 Edit roles&tasks information **3**
 Edit users responsible and modifiers **4**

Task's relations:

Task	Relation
write_user_story	optionalInput
define_release_plan	optionalInput
develop_xp_vision	output

Edit roles&tasks information

Workproduct status:
 Percentage: 0
 User that established it: none
 Edit status of workproduct **5**

Delete Workproduct


You can edit these options of your workproduct

As you can see in the previous image, there are several features you can edit in your workproduct:

- Edit general information: you can edit the name and the description of your workproduct. You're not able to edit the project your workproduct belongs.
- Create template information: if there's no template associated to your workproduct, create a new one using this button.
- Edit role&tasks information: edit what roles are associated to this workproduct responsible and modifiers positions. You can also edit the connections of this workproduct with the tasks that include it.
- Edit users responsible and modifiers: choose what users will work in this workproduct according to the data you set in the workproduct role information.
- Edit status of workproduct: if you are working in this workproduct, you will be able to modify the status of this workproduct. Remember that a log will register your changes in the workproduct status edition. See [Update Status of a Workproduct](#) for further information.

These are all the edition possibilities that are in the workproduct info page. If you want to see further information about the workproduct related information, please go to "[creating a workproduct](#)" section.

Editing your templates

To edit a template, you firstly have to find it in your containing project. Go to your project info page and press the "Templates" tab to open it. Look for the template you want to edit and press the  button next to it.

Once you've press the edit button, you will be redirected to the template edition page. In this page you can edit several things:

Choose the workproduct where you want the template to be. If you choose a workproduct that already has another template you want be able to edit your template. Once you've chosen the workproduct of your template, fill the name and description fields. If you want to upload a new file for your template, mark the "overwrite" checkbox and choose a file from your system to stands for your template. If you don't want to overwrite the existing file in the template, just don't mark this checkbox and finish editing your template.

Editing your project general info and possible users

If you want to edit your project general info and the users that can work in it, you have to go to your project main info page.

As you can see, in the center of this page there's a yellow box where all the EPF info in contained. To edit this information, press the "EPF project info" button. You will enter in a small wizard where you can set the EPF propierties of your project in only 2 steps:


- 1st Step: in this step you can set the Guideline Link of your project (where the extra documentation is hosted), the Subversion Repository Path of your project or the data of the plugin of your project (name, authors and description).
- 2nd Step. Users configuration: here you can set what users will be able to work in your project. Choose them from the drop down box and add them into your project.

When you've edited this data, press the "Finish EPF Editing" button to commit your changes. Now, in the EPF box of your project main info page, you will see that the changes you made have taken effect.

Everything changes: I need to delete information.

A very common operation you will probably want to do in your project is deleting its elements. FVE Project manager has methods to do this in a very simple way. Just keep reading to learn how to delete all the different elements you'll find in your projects.


For each of them, you will always receive a confirmation page to ensure that you really want to delete the element.

 Are you sure that you want to delete this workproduct and his relations with tasks, roles and users?


Delete confirmation page

Remember that once you've pressed the "Delete" button you won't be able to undo your changes.

Deleting your roles

To delete a role, go to a place where it is listed, like in the "EPF Roles" tab in your project main info page. Look for the role you want to delete and press the  button.

Delete role SpokesMan Hablador ?

 Are you really sure to delete this role?

You cannot delete this role because there are tasks or workproducts that depend on it. (See bellow for more info)

Name:	SpokesMan Hablador
Description:	This role has many functions in order to communicate people: He should communicate with the teacher, with the project members, and even with members of other groups to take decisions, announce news, discover anything relevant, etc.
Project:	Avril Lavigne

Remember that the following users play this role:

User
Paco admin

Dependencies


Here you can see all the dependencies of this role. These dependencies can be found both in tasks like workproducts.

Tasks


You cannot delete a role if it has dependencies on it

You will go to the delete confirmation window, where you have to confirm if you really want to delete this role. To avoid inconsistencies in FVE, you may not delete project if they have users associated to them who are working in a specif task o workproduct. If this happens, you have to delete first this association and then try to delete again the role. You can see these dependencies in the same delete confirmation window, in the lowest zone.

Deleting your tasks


Deleting a task is very simple. Just go to the task information page. In the upper-right part of the screen you'll see the next link: " **delete task**". If you press it, a pop up window will ask you if you're really sure to delete this task. If you accept, the task will have been deleted.

Deleting your workproducts

To delete a project, you have to go to a FVE page where it's listed. You can go to the main info page of the project that contains it and open the "Workproduct" tab. Look for the workproduct you want to delete, and press the  button that is placed next to the chosen workproduct.

A confirmation page will appear to make you confirm if you really want to delete the workproduct. You can see the main information of this workproduct. If you're really sure to delete this workproduct, press the "Delete" button and it will be deleted from the system.

Deleting your templates

To delete a template, you have to go to a FVE page where it's listed. You can go to the main info page of the project that contains it and open the "Templates" tab. Look for the template you want to delete, and press the  button that is placed next to the chosen template.

A confirmation page will appear to make you confirm if you really want to delete the template. You can see the main information of this template. Remember that the associated files to this template will be also deleted. If you're really sure to delete this template, press the "Delete" button and it will be deleted from the system.

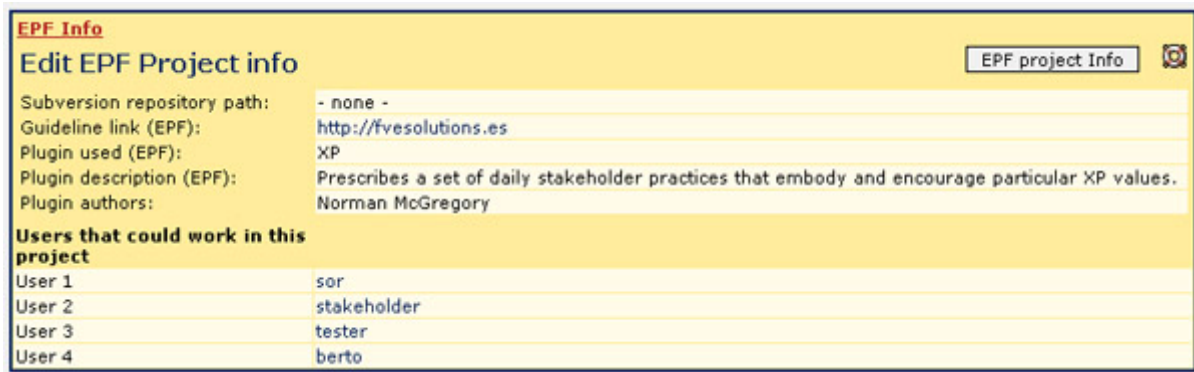
Daily working: I need to see all information.

When your working in a project, you always want to know all the information concerning to it. To obtain this info, you have to consult the show pages of the elements you want to know further information.

In the next sections you will be helped in how to get the information about the elements of FVE Project Manager.

Showing project general info and possible users

To see the information related with the extended EPF information of your project, more specifically, the general EPF info and the users that are able to work in the project you only have to go to the project main info page. In the center of the screen there is a yellow box that shows all this information as you can see in the next image:



EPF Info	
Edit EPF Project info	
Subversion repository path:	- none -
Guideline link (EPF):	http://fvesolutions.es
Plugin used (EPF):	XP
Plugin description (EPF):	Prescribes a set of daily stakeholder practices that embody and encourage particular XP values.
Plugin authors:	Norman McGregor
Users that could work in this project	
User 1	sor
User 2	stakeholder
User 3	tester
User 4	berto

In this box you'll find the EPF info of your projects

Showing project roles

If you want to see the roles that take part of your project, go to the main info page of your project and click on the "EPF Roles" tab. All the roles inserted in your project will appear in this tab.

Showing project tasks

If you want to see the tasks that take part of your project, go to the main info page of your project and click on the "Tasks" tab. All the tasks existing in your project will appear in this tab.



Tabs inside the project info page

Showing project workproducts

If you want to see the workproducts that take part of your project, go to the main info page of your project and click on the "Workproducts" tab. All the workproducts inserted in your project will appear in this tab.

Tasks	Tasks (Inactive)	Forums	Gantt Chart	Task Logs	Events	Files	EPF Roles	Workproducts	Templates	Subversion
Workproducts List of all workproducts that are part of this project (Project Development). <input type="button" value="Create Workproduct"/>										
Operations	Id	Workproduct Name	Brief description					Status		
	11	PlanProject	Plan Project is the main document that specifies how will the project team be organized.					60 %		
	12	OnePage Idea	OnePageIdea is a document that let a user propose an idea to the rest of the workers, specifying main requirements and goals in less than one page.					25 %		
	13	CRC Cards	A CRC card is an index card that is use to represent the responsibilities of classes and the interaction between the classes.The cards are created through scenarios, based on the system requirements, that model the conceptual view of the system.					85 %		
	14	Estimation	This document should estimate software development time according to available resources, risks list, Case-Uses document and Non Functional requirements. This document should use known estimation methods as Parkinson Law, Bailey-Basili model, etc.					0 %		
	15	User Manual	It is the information that is proportionated to the final user in order to know how to use the program. It should be enough explicative to understand all functions of the program without a instructor.					0 %		
	16	Formal Technique Revisions	The members of projects should have frequently reunions to discuss some aspects pof project. Technique formal revision should be the document that show what members participated in the reunion, what aspects has been revised, and what decisions should be communicated to project manager.					0 %		
	17	Concrete details of domain	This document should include everything that is unknown of the domain in order to introduce programmers and software workers in th domain knowledge. Also this document shoud have information about other similar applications in this domain, formats of represent this concrete type of information, etc.					10 %		
	18	Stories	Histories are the first document after "onePageIdea", that defines possible uses of application and the needs that the application shoul solutionate.					0 %		
	19	Technologies Report	It's a document that studies the technologies implicated in the project and establishes advantages and disadvantages of using them.					0 %		

Workproduct tab contents

In the image above you can see that it's very easy to see the overall condition of the workproducts, because of the color code used for it status field. From this tab you can see further information about any workproduct by only clicking on its id number.

Showing project templates

If you want to see the templates that take part of your project, go to the main info page of your project and click on the "Templates" tab. All the templates inserted in your project will appear in this tab.

You'll see a template list where you can find:

- Template name: the name by which the template is know in your project.
- Template description: a small description which can detail the template aim.
- Template workproduct: the workproduct that the template belongs to.

Showing info of a concrete role

To see full information about a single role, you have to go click over its name everywhere in the application it appears. If you know the project what it belongs to, go to the main info page of this project, press the "EPF Roles" tab and look for the role you want to know more information. When you've found it, click on its name and you'll jump to the role info page.

Show information about role Analyst

This is specific information about this role and his relations with tasks, workproducts and users.

General information:		
Id:	19	
Name:	Analyst	
Description:	The analyst in this project has high level functions in software development as discuss and define design, case-uses, etc in the application. The best difference between a programmer is more responsibility in design and less in code implementation matters.	
Part of project:	Avril Lavigne	
Workproduct's relations:		
Workproduct 1. Name:	Software Requirements Specification	Responsible
Workproduct 2. Name:	UseCases Document	Responsible
Workproduct 3. Name:	Functional Requirements	Responsible
Workproduct 4. Name:	UML Design	Responsible
Workproduct 5. Name:	Design Explanation	Responsible
Edit associated workproducts responsible		
Doesn't have Workproducts as modifiers		
Edit associated workproducts modifiers		
Task's relations:		
Task 1. Name:	Propose functional requirements	Primary
Task 2. Name:	Create Case Uses	Primary
Task 3. Name:	Generate a first design with classes	Primary
Edit associated tasks primary info.		
Task 1. Name:	Make a brainstorming	Additional
Task 2. Name:	Get support for ideas	Additional
Task 3. Name:	Discuss the more interesting ideas	Additional
Task 4. Name:	Vote ideas	Additional
Task 5. Name:	Imagine situations in which the application could be useful	Additional
Task 6. Name:	Writing histories	Additional
Task 7. Name:	Contract meetings with other members	Additional

Role information page

The information you'll see about the role is divided into 4 parts:

- General information: it's the role main data. There is information about the role name, description or the project that it belongs to.
- Workproduct relations: these are the workproduct in which the role has participation. The role can participate as a responsible (main user in the workproduct) or modifier (users that can modify this workproduct).
- Task relations: these are the tasks in which the role participate. This participation can be primary (the main user that works in the task) or addition (the user participate in the task but cannot modify it).
- Users info: here you'll find the users that can play this role. Remember that the role is associated to a unique project, so these users will only be

able to play this role in the project the role belongs to.

Showing info of a concrete task

To see full information about a single task, you have to go click over its name everywhere in the application it appears. If you know the project what it belongs, go to the main info page of this project, press the "Tasks" tab and look for the task you want to know more information. When you've found it, click on its name and you'll jump to the task info page.

In this info page, you will find the next information:

Details

Project: **Project Development**

Task: **Make a brainstorming**

Task Parent: Propose a idea for a project

Owner: admin

Priority: normal

Web Address:

Milestone: No

Progress: 0%

Time Worked: 0

Dates and Targets

Start Date: 22/04/2008 10:00 am

Finish Date: 27/04/2008 11:00 am

Expected Duration: 1 hours

Target Budget \$: 0.00

Task Type : Unknown

Assigned Users

Pedro Pérez | pedro.perez@rb.es

Dependencies

none

Tasks depending on this Task

none

Description

Everyone can contribute ideas, for extrain that is seems at principle. This process usually gives as result many useless info, but also one or a bit very brilliant ideas.

EPF Info

Edit EPF Task info

Task Type

Pues una tarea

Additional Users

none

Responsible User

Paco SpokesMan Hablador

Associated Workproducts

Name	TypeDependency	Status	Revised by
PlanProject	OptionalInput	60 %	admin
OnePage Idea	output	25 %	admin

Workproduct "Status" Key: ■ = 0 - 20% ■ = 20 - 40% ■ = 40 - 60% ■ = 60 - 80% ■ = 80 - 100% ■ = Finished

Workproduct "Revised by" Key: ■ = None ■ = Any ■ = Modifier user ■ = Responsible user

Task information page

As you can see in the image above, you will find information about:

- Task main information: this is the default information linked to the task (number 1 in the image). You will see its start and finish dates, the project that this task belong or the task main description.
- Taks EPF Information: FVE Project Manager inserts this new information to the projects tasks. As you can see, there is information about the task type (2), the additional users that work on it (3), the responsible and unique user in charge of this task (4) and the

workproducts that are attached to the task (5). These workproducts have a color code to help you seeing at a glance what the current status of these workproducts is and who was the last user in making a change on them.

Showing info of a concrete workproduct

To see full information about a single workproduct, you have to go click over its name everywhere in the application it appears. If you know the project what it belongs, go to the main info page of this project, press the "Workproducts" tab and look for the workproduct you want to know more information. When you've found it, click on its name and you'll jump to the workproduct info page.

Show information about workproduct PlanProject

This is specific information about this workproduct and his relations with tasks, roles and users.

General information:

Id: 11

Name: PlanProject

Description: Plan Project is the main document that specifies how will the project team be organized.

Part of project: Avril Lavigne

[Edit general information](#)

Template associated:

Doesn't have Associated Template

[Create template information](#)

Role's relations:

User responsible: admin

Role responsible: Project Manager

Doesn't have Modifiers Users And Modifiers Roles

[Edit roles&tasks information](#)

[Edit users responsible and modifiers](#)

Task's relations:

Task	Relation
Update Plan Project	mandatoryInput
Update Plan Project	output
Make a Plan Project and Iteration Plan	mandatoryInput
Make a Plan Project and Iteration Plan	output
Organize work in order to member skills	output
Make first the high priority use cases	mandatoryInput
Implements second the medium priority use cases	mandatoryInput

Workproduct information page

In this info page you will find the next workproduct information:

- General information: The main information of this workproduct as its name, description or the project the workproduct belongs to.
- Template associated: this field indicates if there's a template linked to this workproduct. If so, press its name to see additional information.
- Role's relations: here you will find information about the users who

participate in this workproduct and what role do the play.

- Task's relations: workproducts are associated to one or more tasks. In this field you can see what tasks are related with this workproduct and what kind on relation exists between them.
- Workproduct status: users who participate in the workproduct may modify the workproduct completion status. In this field you will see what the current status of the workproduct is and who was the last user in modifying it.

Showing info of a concrete template

To see full information about a single template, you have to go click over its name everywhere in the application it appears. If you know the project what it belongs, go to the main info page of this project, press the "Templates" tab and look for the task you want to know more information. When you've found it, click on its name and you'll jump to the template info page.

Id:	8
Name:	Word Doc
Description:	Word document which contents all the information about the use cases.
File:	Press here to link to the file or right click to download it
Part of project:	Project Development
Workproduct:	Limoncello

Template information page

Inside this info page, if you want to dowload file associated to the template, just press the "Press here to link to the file or right click to download it" link.

Showing possible roles of a user

If you want to know what roles can play a specific user, you have to go to its user info page. To go to this screen, you have to click on the user name wherever it appears in FVE. For example, it the EPF info of a project you can see what users are working in it.

Remember that you have to be an administrator to see the info page of a user who is different from the user you're logged in with.

Once you're in the user info page, just press the "EPF Possible Roles" tab to see what roles can play this user.



Tabs in the user info page: General.

This tab contains all the roles the user can play or be organized by the project they belong to. You can also find the description of the role next to its name.

Showing specific roles of a user

If you want to know what roles are currently playing a specific user, you have to go to its user info page. To go to this screen, you have to click on the user name wherever it appears in FVE. For example, in the EPF info of a project you can see what users are working in it.

Remember that you have to be an administrator to see the info page of a user who is different from the user you're logged in with.

Once you're in the user info page, just press the "EPF Specific Roles" tab to see what roles can play this user.

Projects

Projects gantt

User Log

Permissions

Roles

EPF Possible Roles

EPF Specific Roles

EPF Tasks

EPF Workproducts

Roles

List of all roles that this user is currently playing in every project.

Project Development0%

Order by	Role name	Role Description
	QAInspector	His role should define and try to achieve success in QA practices. Also this role should control that everything goes as has been planned.

Empezamos de nuevo0%

Order by	Role name	Role Description
	tester	This role is responsible for the core activities of the test effort. Those activities include identifying, defining, implementing, and conducting the necessary tests, as well as logging the outcomes of the testing and analyzing the results.
	developer	This role is responsible for developing a part of the system, including designing it to fit into the architecture, possibly prototyping the user-interface, and then implementing, unit-testing, and integrating

Tabs in the user info page: EPF Specific Roles.

As you can see in the image above, the user specific roles will be listed ordered by the project they belong to. You can also find the description of the role next to its name.

Showing tasks of a user (as primary or additional performer)

If you want to know in what tasks is currently working a specific user, you have to go to its user info page. To go to this screen, you have to click on the user name wherever it appears in FVE. For example, in the EPF info of a project you can see what users are working in it.

Remember that you have to be an administrator to see the info page of a user who is different from the user you're logged in with.

Projects	Projects gantt	User Log	Permissions	Roles	EPF Possible Roles	EPF Specific Roles	EPF Tasks	EPF Workproducts
----------	----------------	----------	-------------	-------	--------------------	--------------------	-----------	------------------

Tasks

List of all tasks that has this user as responsible or as additional performer in every project.

:: Project Development 0%

Order by	Task name	As role	Task Relation
	Elaborate standards of code and documentation	QAInspector	PrimaryPerformer
	Take decisions about configuration management system	QAInspector	PrimaryPerformer
	Makes an estimation of the size and time to develop the project	QAInspector	PrimaryPerformer
	Actualize	QAInspector	PrimaryPerformer
	Generate stats of group working	QAInspector	PrimaryPerformer

:: Empezamos de nuevo 0%

Order by	Task name	As role	Task Relation
	implement_test_scripts	tester	PrimaryPerformer
	design_solution	developer	PrimaryPerformer
	run_developer_tests	developer	PrimaryPerformer
	implement_developer_tests	developer	PrimaryPerformer

Tabs in the user info page: EPF Tasks.

Once you're in the user info page, just press the "EPF Tasks" tab to see in what tasks is the user currently working. For each task you'll see what role is the user playing and what stand plays the user in the task with that role: primary or additional performer.

Showing workproducts of a user (as responsible or additional performer)

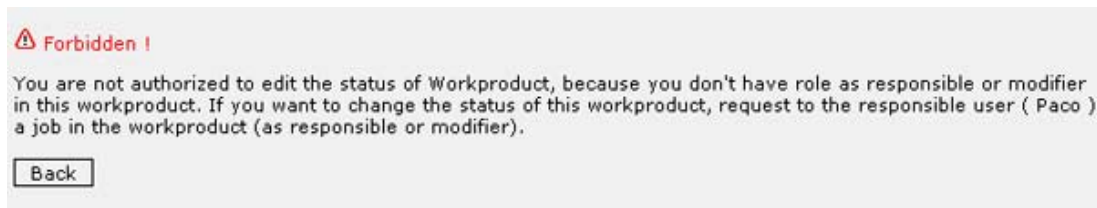
If you want to know in what workproducts is currently working a specific user, you have to go to its user info page. To go to this screen, you have to click on the user name wherever it appears in FVE. For example, in the EPF info of a project you can see what users are working in it.

Remember that you have to be an administrator to see the info page of a user who is different from the user you're logged in with.

Once you're in the user info page, just press the "EPF Workproducts" tab to see in what workproducts is the user currently working. For each workproduct you'll see what role is the user playing and what stand plays the user in the workproduct with that role: responsible or additional performer.

Update status of a Workproduct

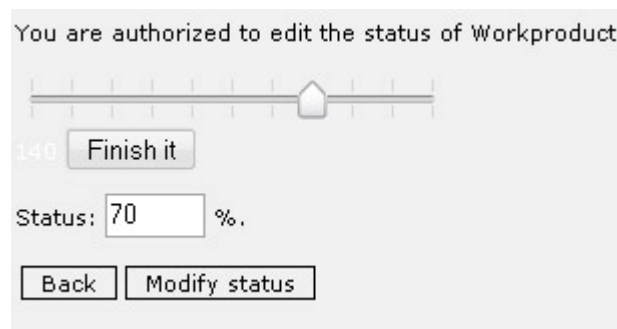
The workproduct status indicates how finished the workproduct is. So, if you take part in that workproduct (as a responsible or a modifier user) you are able to modify the workproduct status. If you have no participation in the workproduct, you will be informed with the next screen:



Workproduct: You have no permission to edit this workproduct status

You only can inform you administrator in order to make him registering you in the workproduct working users. Once the admin has set this, you will be able to edit the workproduct status.

To update the status of a workproduct, first you have to go to its info page ([showing info of a concrete workproduct](#)). In this page, you will see in the lowest part of the form, a link that says "edit status of workproduct". If you press this link, you will go to the status editing page.



Modifying your workproduct status

At this point, you have to choose the workproduct completion status. Use the sliding bar to select a value from 0 to 100 %. You can also press the "Finish it" button to indicate the system that the workproduct is finish (100%). Once you're ready, press the "Modify Status" to confirm the changes. Remember that your user will be logged as it was the last user that made a change in the workproduct status.

Reports: You can use them as usual standards reports

An advantage of FVE Manager is that in your new FVE projects you can also make reports about the information contained in these projects.

You have several different reports that you can apply to your projects to help you make a summary of the status of the project. To access to these reports, you have to go to the project main info page and press the "reports" link at the top of the page.

projects list : edit this project : organize tasks : **reports** delete project

Avril Lavigne

Details	Summary
Company: ska	Status: Not Defined
Short Name:	Priority: normal
Start Date: 02/06/2008	Type: Unknown
Target End Date: 02/06/2008	Progress: 0.0%
Actual End Date: 27/06/2008	Worked Hours:
Target Budget: \$0.00	Scheduled Hours: 62
Project Owner: Person, Admin	Project Hours: 62
URL:	
Staging URL:	
Description	
Canta muy bien	

EPF Info Edit EPF Project info EPF project Info

Reports link to access to the reports list

You'll be sent to the reports page, where you have to choose the report you want to apply to your project.

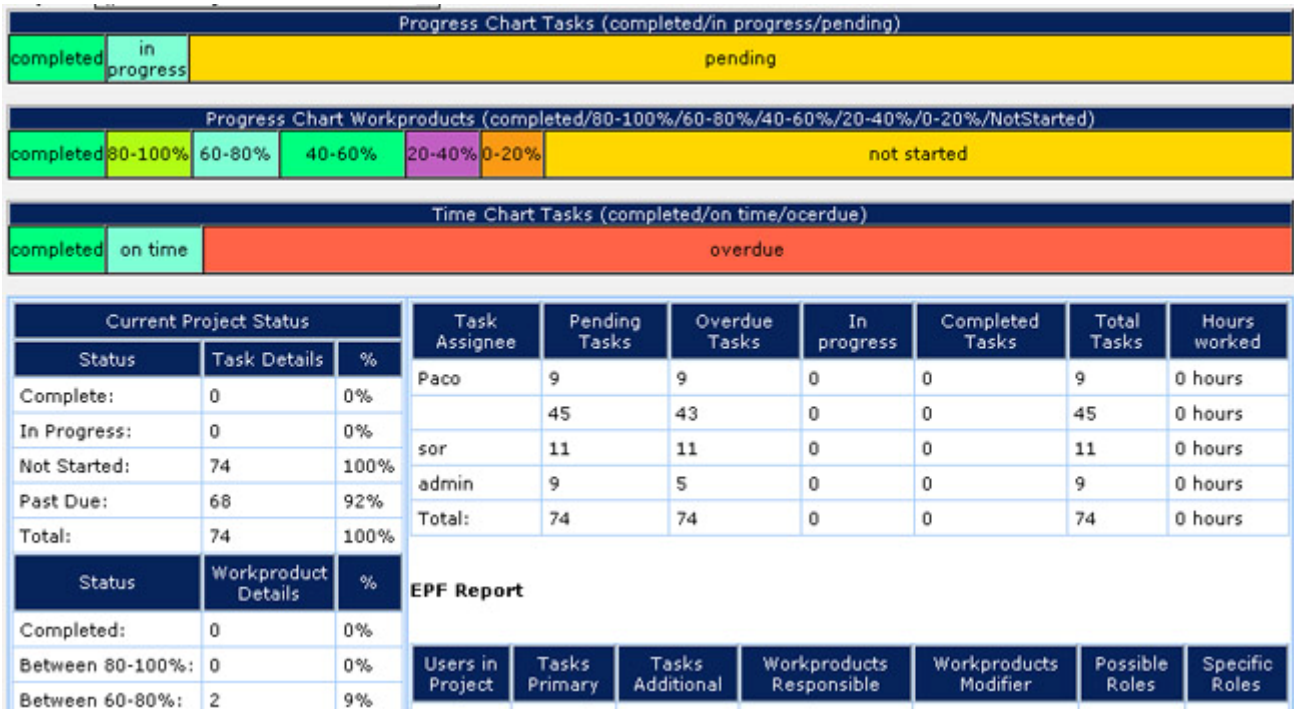
Reports: New Project Statics with FVE information

There is a new report in the report list that shows the statistics of the projects including the EPF elements that FVE Project Manager includes.

Reports Available	
Allocated user hours	- This report shows how many hours a user has allocated in the specified period of time.
Completed	- Tasks completed in the last week
Overall Report	- Gives an overall snapshot of all projects
Overdue	- Tasks currently overdue
Project Statistics	- Project Statistics, describing task statuses.
Tasks end date	- View the task's end date (real and theoretical)
Task List	- View the Projects Task List
Task Log	- View the user task logs
Tasks sorted by user	- This report shows the tasks allocated to each user.
Upcoming	- Tasks to be completed in next week
User performance	- Report that shows the amount of hours worked by a user against those allocated in tasks
Project Statistics: Include FVE Stats	- Project Statistics, describing task and workproducts statuses and info of FVE per user.

New FVE Report

The name of this new report is "Project Statistics: include FVE Stats". You can access it in the same way you access the rest of the reports.



New FVE Project Manager report

In this new report you can see statistics about:

- Workproduct completion chart
- Table with workproducts completion percentages.
- Task assignee tables
- Tables with users and EPF elements relationships.

This new report will help users to get a complete idea of the current status of the project.

G.4. Subversion

In this chapter you will see how fve makes subversion repositories management on dotProject. Concretely fve interacts as a linker, model viewer and light subversion operations in all containing aspects.

1. Introduction to fve subversion features

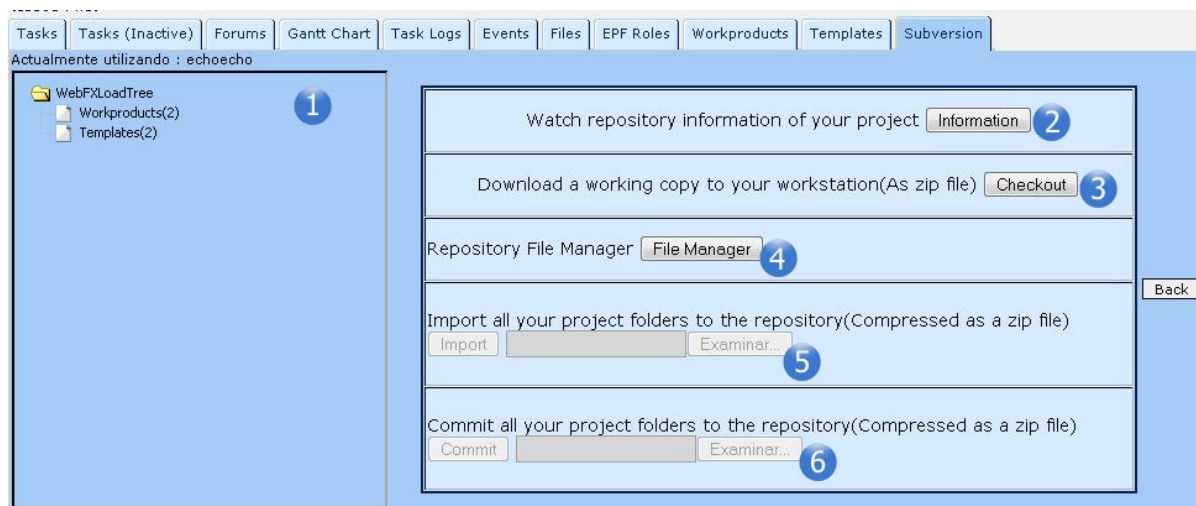
Fve module is a concrete linker between projects created in dotProject and their physical contents. To do that, fve manage a few tables in a database where all relative information of Subversion is stored (mostly paths and permissions). At this point we have already notice that fve is divided in two parts: The first one is the fve php module where the most part of the application lies, and the second one, implemented in javaEE as a

war module that contains java svnkit libraries who makes Subversion operations possible.

The subversion operations considered in fve are all the basic ones: to add and remove files and directories from a repository. From fve interfaces you can also download checkouts and make general commits, but this use is not quite recommended for the massive data storage problems. Those operations may be considered in lightweight data projects. To do these operations fve provides an easy interface in dotproject that will introduce later.

2. Subversion interface in fve

The main subversion interface view it's an dotproject interface that can be access via Subversion Tab in dotproject 'Project view'.



Subversion Tab Interface in dotproject

The interface is separated in two parts:

1. The repository view tree: Is the faster way to view the actual status of the project as a file tree view.
2. The main menu option list: The list of options is described below (numbers 2 to 5):

2) Information from a project

This option shows the common subversion information tips associattted with a project repository. Its uses are frequent for evaluating continous work on it with revision processes.



Subversion project information page

3) Checkout Option

This option generates a zip file from the working copy to download the entire project at once. Due to online restrictions of download multiple files, this solution makes us guarantee that we get all the files and folders from an unique download. (This option has problems to work if the apache server has limited virtual space or the repositories are too big, so check it).

4) File Manager

The file manager is the core of the subversion feature. It lets you work with a remote working copy that commits and updates automaticly every change you make. An example of how the interface of the manager looks like is as shown below. The explanation of each of the file manager component it's also described below the image.



Subversion FileManager Interface

Read options and navigation through the repository issues

1. Name and relative path information

In this area you will see the name of the project you are in, and the relative path that the "File Manager" is showing actually.

2. Parent Folder

To go to the previous folder (if we are inside another) you have to press the up folder icon.

3. Folder explorer

Pressing a folder name or icon makes you navigate over the repository accessing the selected folder. It's mostly like the Windows Explorer.

4. File explorer

Pressing a file name or icon makes you open the file with your internet explorer. If you do this you should know that only recognized file extensions for your internet explorer will produce the correct display of the file. For the rest of them you should use the 'download link as' option that every internet explorer provides. This will download the requested file to your computer.

Writing changes into the repository

5. File upload

The file upload option button is used to introduce new files into the repository, file manager upload the file submitted into the online working copy and commit the changes directly to the repository.

6. Create new folders

With the create folder option button, file manager will create a folder into the online working copy with the name submitted and commit the changes directly to the repository.

7. Delete archives

Every folder and file can be deleted from the manager pressing his red cross associated icon. The changes will appear in the repository as in the online working copy.

As we see in the next image, every project created with fve contains default templates and workproducts paths where those files are stored. For security consistance with the databases of fve, those folders are just related for seen its contents and not to modify them. To do that we need to add or remove them from the Workproduct or Template Tabs in dotproject project view.



Initial contents of a subversion project (Showing FileManager Interface)

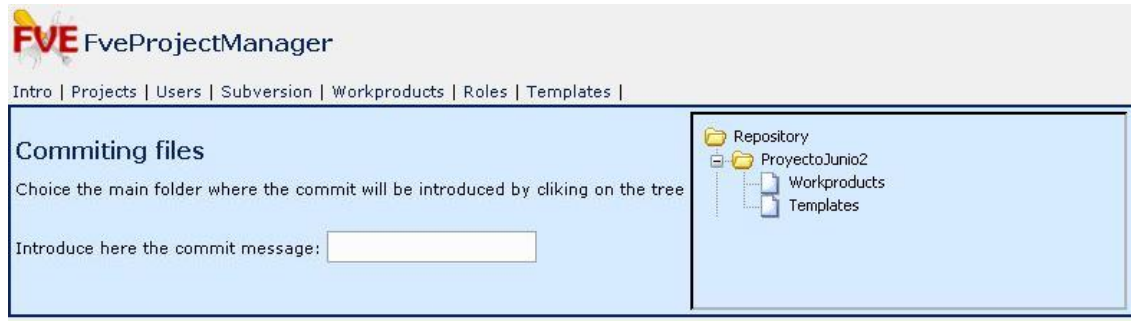
5) Import Option

This option lets you specify a zip file with your project to upload to the repository, even there aren't previous versions of the files of the project, if you click in the option "Import" and select your zip file.

6) Commit Option

This option allows the user to upload multiple files for commit into the repository in two easy steps.

First of all we have to compress all our files into a zip file. Into the subversion menu panel we browse the zipped file and step. After uploading the file, a new window will appear with this look:



Commit files Subversion Interface

In this window once you have introduced a commit message (optional) you have to press on the repository tree on one of the items. This tree represents only the folders over your repository, clicking on one of them will commit the uploaded files under it. If folders and files that the repository have before the commit matches with the uploaded files, those will be committed and replaced.

3. Other subversion links over dotproject

Integration of subversion resides also in other fve interfaces. An example of it is the possibility of accessing the template files from the repository from the templates view (The same happens in workproducts).

4. Subversion permissions strenght in fve

Normally, when you try to connect to a subversion repository, you have to introduce your user and password to authenticate yourself. Fve does this work for you using your registered user and subversion password to authenticate you. There are two ways where fve acts differently:

1. Working in local: When a local repository is used, the security files from svn server (named as authz and passwd) are managed by fve php module, and before riching any local subversion operation (with all permits access), fve php will check them and block you if necessary. Using these files allow us to deploy an svnserver (svn server service) that can be access from outside of fve using subversion with an url path begining by "svn:/". For more information about deploying svnserver click [here](#)
2. Working with an external repository: When an external repository is set for your project, fve keeps you subversion user and pass to

access it. Those user and pass are normally different from the dotproject user and pass. They will be used in every subversion operation by svnkit libraries (fve javaEE package) to guarantee the correct access. (Under development).

G.5. Glossary

The glossary contains all the terms that are used in the application. You will find a particular description for each of them to help you understanding them.

Term	Description	References		
Workproduct	Workproduct is a general term for task inputs and outputs, descriptions of content elements that are used to define anything used, produced, or modified by a task.	EPF Documentation	Process	Framework
EPF Role	A role is a well-defined set of related skills, competencies, and responsibilities. Roles can be filled by one person or multiple people. One person may fill several roles. Roles perform tasks.	EPF Documentation	Process	Framework
Template	A template is a specific type of guidance that provides a work product with a predefined table of contents, sections, packages, and headings. Templates provide a standardized format, as well as descriptions of how the sections and packages are supposed to be used and completed. Templates can be provided for documents as well as conceptual models or physical data stores.	EPF Documentation	Process	Framework
EPF Task	A task is an assignable unit of work. Every task is assigned to a specific role. The duration of a task is generally a few hours to a few days. Tasks usually generate one or more work products. This type of task is called EPF Task because is the Eclipse Process Framework information assigned to tasks.	EPF Documentation	Process	Framework
Delivery Proccess	A delivery process is the process that covers a whole development lifecycle from beginning to end. A delivery process can be used as a template for planning and running a project. It provides a complete lifecycle model with predefined phases, iterations, and activities.	EPF Documentation	Process	Framework
Capability Pattern	Capability patterns are a special type of process that describes a reusable cluster of activities in common process areas. Capability patterns express and communicate process knowledge for a key area of interest such as a discipline and can be directly used by a process practitioner to guide his work. Capability patterns are also used as building blocks to assemble delivery processes or larger capability patterns ensuring optimal reuse and application of the key practices they express. Examples of capability patterns include "use case-based requirements management", "use case analysis", or "unit testing". Typically, but not necessarily, capability patterns have the scope of one discipline providing a breakdown of reusable complex activities, relationships to the roles which perform tasks within these activities, as well as to the work products that are used and produced. A capability pattern does not relate to any specific phase or iteration of a development lifecycle, and should not imply any. In other words, a pattern should be designed in a	EPF Documentation	Process	Framework

way that it is applicable anywhere in a delivery process, thereby enabling its activities to be flexibly assigned to whatever phases there are in the delivery process to which it is being applied.

When developing a process, it is not necessary to develop the process from scratch, adding descriptors one by one. You can reuse existing capability patterns or even capability pattern parts to individually customize the pattern's content to the particular situation for which it is applied. A capability pattern must be applied to one specific activity in a process.

Activities are the main building blocks for processes. An activity is a collection of work breakdown elements such as task descriptors, role descriptors, work product descriptors, and milestone. Activities can include other activities.

Activity	Activities can be presented in work breakdown structures and activity diagrams that graphically describe the flow of work by showing which activities precede other activities. phase and iteration are special types of activities that define specific properties.	EPF Process Documentation	Framework
Phase	A phase is a type of activity that represents a significant period in a project. Phases typically conclude with a management checkpoint, milestone or set of deliverable artifacts.	EPF Process Documentation	Framework
Iteration	An iteration is a group of activities that are repeated more than once. Iterations are used to organize work into repetitive cycles.	EPF Process Documentation	Framework